ARMY RESEARCH LABORATORY

The Anticipative Maximin Adaptive-Array Algorithm for Frequency-Hopping Systems

by Don Torrieri

ARL-TR-3777 **April 2006**

NOTICES

Disclaimers

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof.

Destroy this report when it is no longer needed. Do not return it to the originator.

Army Research Laboratory
Adelphi, MD 20783-1197

ARL-TR-3777 **April 2006**

The Anticipative Maximin Adaptive-Array Algorithm for Frequency-Hopping Systems

Don Torrieri
Computational and Information Sciences Directorate, ARL

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED (From - To)
April 2006	Final	October 2005 to present

4. TITLE AND SUBTITLE
The Anticipative Maximin Adaptive-Array Algorithm for Frequency-Hopping Systems

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Don Torrieri

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
U.S. Army Research Laboratory
ATTN: AMSRD-ARL-CI
2800 Powder Mill Road
Adelphi, MD 20783-1197

8. PERFORMING ORGANIZATION REPORT NUMBER
ARL-TR-3777

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
U.S. Army Research Laboratory
2800 Powder Mill Road
Adelphi, MD 20783-1197

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT
The *anticipative maximin algorithm* is a blind adaptive-array algorithm that exploits both the spectral and temporal characteristics of frequency-hopping signals. The anticipative maximin algorithm fuses a new form of anticipative processing with an improved version of the maximin algorithm to enable the cancellation of partial-band interference within a hopping band. Major advantages are that the anticipative maximin algorithm requires neither training sequences, directional information, nor decision-directed adaptation. The algorithm provides frequency compensation when the hopping spans a wide spectral band. The algorithm is derived and shown to converge. Details of the design of the anticipative maximin processor are presented. Extensive simulation experiments confirm the effectiveness of the algorithm against partial-band interference under a variety of scenarios.

15. SUBJECT TERMS
Adaptive array, frequency hopping, interference suppression

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE	UL	55	Don Torrieri
Unclassified	Unclassified	Unclassified			19b. TELEPHONE NUMBER (*Include area code*) (301) 394-2484

Standard Form 298 (Rev. 8/98)
Prescribed by ANSI Std. Z39.18

Contents

List of Figures

List of Tables

1. Introduction

Frequency hopping and adaptive arrays are two of the most powerful methods of rejecting interference. However, the multitude of standard adaptive algorithms are not entirely compatible with frequency hopping, and more specialized algorithms are required. An early attempt to use the classical *least-mean-squares (LMS) algorithm* (*1*) placed restrictions on the hopping and did not exploit the specific characteristics of the frequency-hopping waveform. The two blind adaptive-array algorithms that do exploit the characteristics are the *maximin algorithm* (*2*), (*3*) and the *constant power algorithm* (*4*). Since it exploits the constant power of the frequency-hopping waveform, the constant power algorithm is inherently suitable only over the flat-fading channel and against strong interference signals that fluctuate significantly from hop-to-hop.

The *maximin algorithm* discriminates between the desired signal and interference on the basis of the distinct spectral characteristics of spread-spectrum signals. As indicated by its name, the maximin algorithm simultaneously maximizes the desired-signal component and minimizes the interference component in the despread signal. An adaptive array using the maximin algorithm provides a frequency-hopping system with a high degree of protection against strong directional interference that occupies a large part of the hopping band. Major advantages of the maximin algorithm are that it does not require training sequences, directional information, or decision-directed adaptation.

The *anticipative maximin algorithm* is a blind adaptive-array algorithm that exploits both the spectral and temporal characteristics of frequency-hopping signals. The anticipative maximin algorithm fuses a new form of anticipative processing with an improved version of the maximin algorithm to enable the cancellation of partial-band interference within a hopping band. Anticipative processing was originally proposed to compensate for the impairments of very wide hopping bands (*5*). The improvements of the maximin algorithm include a new adaptation sequence, the introduction of a memory factor, and a simplification of the baseband signal processing.

The next section introduces the notation and provides a derivation of the improved maximin algorithm. In section 3, the details of the implementation of the maximin processor in a frequency-hopping system are explained. The convergence analysis of appendix B establishes bounds on the adaptation constant and justifies the adaptation sequence used in the maximin algorithm. The full anticipative maximin algorithm and its implementation are explained in section 4. The results of many representative simulation experiments are presented in section 5. Frequency compensation techniques for wideband frequency hopping are described in section 6.

2. Derivation of Maximin Algorithm

The basic configuration of an adaptive array for frequency-hopping systems is displayed in figure 1. The unmodulated frequency-hopping replica, which is produced by a synchronized receiver (*6*), enables the dehopping or removal of the time-varying carrier frequency of each copy of the received frequency-hopping signal. The sample values of the complex envelopes of the dehopped signals are extracted by initial processors and applied to an adaptive processor that executes the maximin algorithm. The desired signal and the interference are assumed to arrive at an adaptive array of N antennas. The desired signal, interference signals, and thermal noise are modeled as independent zero-mean, wide-sense-stationary stochastic processes. Let $\mathbf{x}(i)$ denote the discrete-time vector of the complex envelopes of the N antenna outputs after each one has been sampled, despread, and filtered. The index i denotes the sample number. The vector $\mathbf{x}(i)$ can be decomposed as

$$\mathbf{x}(i) = \mathbf{s}(i) + \mathbf{n}(i) \tag{1}$$

where $\mathbf{s}(i)$ is the vector of desired-signal complex envelopes, and $\mathbf{n}(i)$ is the vector of interference and thermal-noise complex envelopes. The adaptive filter generates a weight vector $\mathbf{w}$ with complex-valued components. The output of the adaptive filter is

$$y(i) = \mathbf{w}^H \mathbf{x}(i) = y_s(i) + y_n(i) \tag{2}$$

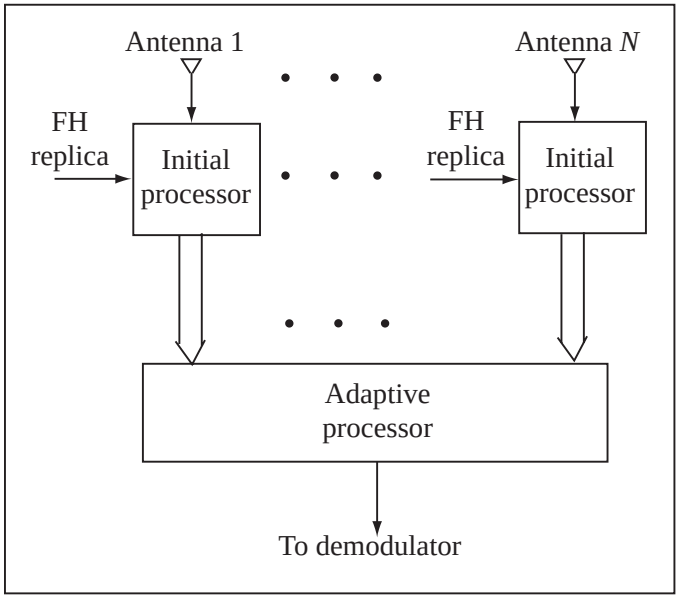

Figure 1. Configuration of adaptive array for frequency-hopping system.

where $\mathbf{M}^H$ denotes the complex-conjugate transpose of $\mathbf{M}$ and

$$y_s(i) = \mathbf{w}^H \mathbf{s}(i), \qquad y_n(i) = \mathbf{w}^H \mathbf{n}(i). \tag{3}$$

Let $E[\ ]$ denote the expected value. When $\mathbf{w}$ is a constant, the desired-signal output power is

$$p_s = E[|y_s(i)|^2] = \mathbf{w}^H \mathbf{R}_s \mathbf{w} \tag{4}$$

where $\mathbf{R}_s$ is the desired-signal correlation matrix:

$$\mathbf{R}_s = E[\mathbf{s}(i)\mathbf{s}^H(i)]. \tag{5}$$

The interference-plus-noise output power is

$$p_n = \mathbf{w}^H \mathbf{R}_n \mathbf{w} \tag{6}$$

where $\mathbf{R}_n$ is the interference-plus-noise correlation matrix:

$$\mathbf{R}_n = E[\mathbf{n}(i)\mathbf{n}^H(i)]. \tag{7}$$

The signal-to-interference-plus-noise ratio (SINR) is

$$\rho = \frac{p_s}{p_n} = \frac{\mathbf{w}^H \mathbf{R}_s \mathbf{w}}{\mathbf{w}^H \mathbf{R}_n \mathbf{w}} \ . \tag{8}$$

The SINR provides the performance measure that the adaptive algorithm seeks to maximize. The optimal weight vector derived from the maximization of the SINR (appendix B) is no different from the optimal weight vector under other optimization criteria such as the minimum-variance distortionless-response criterion (*7*), (8), (*9*). However, the adaptive algorithm obtained from the SINR optimization criterion differs from the algorithms derived under other criteria in that the recursive equation is a highly nonlinear function of the weight vector.

The weight vector may be decomposed as $\mathbf{w} = \mathbf{w}_r + j\mathbf{w}_i$, where $\mathbf{w}_r$ and $\mathbf{w}_i$ are the real and imaginary parts of $\mathbf{w}$, respectively, and $j = \sqrt{-1}$. Let ∇_{wr} and ∇_{wi} denote the gradients with respect to $\mathbf{w}_r$ and $\mathbf{w}_i$, respectively. The complex gradient is defined as

$$\nabla_w = \frac{1}{2}(\nabla_{wr} + j\nabla_{wi}) \ . \tag{9}$$

The maximin algorithm is based on the *method of steepest descent* (*7*), (8), (*9*). In this method, the weight vector is changed along the direction of the negative gradient of a

performance measure that is to be minimized. Separate steepest-descent equations can be written for $\mathbf{w}_r$ and $\mathbf{w}_i$ with the negative SINR serving as the performance measure. Combining these equations yields

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mu_0(k)\nabla_w \rho(k) \tag{10}$$

where k denotes the weight iteration number, $\mu_0(k)$ is a scalar sequence that controls the rate of change of the weight vector, and $\nabla_w \rho(k)$ is the gradient of the SINR at iteration k. Applying equation 8 to 9 and evaluating, we obtain

$$\nabla_w \rho = \rho \left[\frac{\mathbf{R}_s \mathbf{w}}{p_s} - \frac{\mathbf{R}_n \mathbf{w}}{p_n} \right]. \tag{11}$$

Since the interference and noise are zero-mean and statistically independent of the desired signal, the *input correlation matrix* is

$$\mathbf{R}_x = E[\mathbf{x}(i)\mathbf{x}^H(i)] = \mathbf{R}_s + \mathbf{R}_n \tag{12}$$

and the adaptive-filter output power is

$$p_x = E[|y(i)|^2] = \mathbf{w}^H \mathbf{R}_x \mathbf{w} = p_s + p_n. \tag{13}$$

Substitution of equation 12 and 13 into 11 and simplification yields

$$\nabla_w \rho = (\rho + 1) \left[\frac{\mathbf{R}_x \mathbf{w}}{p_x} - \frac{\mathbf{R}_n \mathbf{w}}{p_n} \right]. \tag{14}$$

Substitution of this equation into equation 10 and the replacement of $\mathbf{w}$ with $\mathbf{w}(k)$ gives the *steepest-descent algorithm*:

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mu_0(k)[\rho(k) + 1] \left[\frac{\mathbf{R}_x \mathbf{w}(k)}{p_x(k)} - \frac{\mathbf{R}_n \mathbf{w}(k)}{p_n(k)} \right]. \tag{15}$$

If $\mathbf{w}$ is modeled as deterministic (nonrandom), then $\mathbf{R}_x \mathbf{w}(k) = E[\mathbf{x}(i)y^*(i)]$ and $\mathbf{R}_n \mathbf{w}(k) = E[\mathbf{n}(i)y_n^*(i)]$, where the asterisk denotes the complex conjugation. Thus, we can avoid estimating the matrices $\mathbf{R}_x$ and $\mathbf{R}_n$ by finding estimators of $E[\mathbf{x}(i)y^*(i)]$ and $E[\mathbf{n}(i)y_n^*(i)]$.

A simplification that ultimately reduces the amount of computation by nearly a factor of two is obtained by observing that $\mathbf{x}(i)$ is obtained by sampling a continuous-time vector of complex envelopes. In each array branch, the thermal noise is independent of the noise in

the other branches, and each desired or interference signal is a delayed version of the corresponding signal in the other branches. Therefore, the circular symmetry of the complex envelopes (appendix A) implies that

$$E[\mathbf{x}(i)\mathbf{x}^T(i)] = \mathbf{0} \tag{16}$$

and

$$E[\mathbf{n}(i)\mathbf{n}^T(i)] = \mathbf{0} \tag{17}$$

where $\mathbf{M}^T$ denotes the transpose of $\mathbf{M}$. The adaptive-filter output can be decomposed as

$$y(i) = y_r(i) + jy_i(i) \tag{18}$$

where $y_r(i)$ and $y_i(i)$ are the real and imaginary parts of $y(i)$, respectively. If the weight vector is modeled as deterministic between iterations, then equation 2 and 16 imply that

$$E[\mathbf{x}(i)y_r(i)] = E\left[\mathbf{x}(i)\left\{\frac{1}{2}\mathbf{x}^H(i)\mathbf{w}(k) + \frac{1}{2}\mathbf{x}^T(i)\mathbf{w}^*(k)\right\}\right] = \frac{1}{2}E\left[\mathbf{x}(i)\mathbf{x}^H(i)\right]\mathbf{w}(k) \tag{19}$$

where the index i refers to a sample taken between weight iterations k and $k+1$. This equation and equation 12 yield

$$\mathbf{R}_x\mathbf{w}(k) = 2E[\mathbf{x}(i)y_r(i)]. \tag{20}$$

Similarly,

$$\mathbf{R}_n\mathbf{w}(k) = 2E[\mathbf{n}(i)y_{nr}(i)] \tag{21}$$

where $y_{nr}(i)$ is the real part of $y_n(i)$. Equations 2 and 16 imply that $E[y^2(i)] = 0$, and the substitution of equation 18 yields $E[y_r^2(i)] = E[y_i^2(i)]$ and $E[y_r(i)y_i(i)] = 0$. Therefore, $p_x(k) = E[|y(i)|^2] = 2E[y_r^2(i)]$. Similarly, $p_n(k) = 2E[y_{nr}^2(i)]$.

To derive the maximin algorithm, let $\hat{p}_x(k)$ and $\hat{p}_n(k)$ denote estimates of $E[y_r^2(i)]$ and $E[y_{nr}^2(i)]$, respectively, following weight iteration k. The estimate of ρ following iteration k is $\widehat{\rho}(k)$. Let $\mathbf{c}_x(k)$ and $\mathbf{c}_n(k)$ denote estimates following iteration k of the input correlation vector $E[\mathbf{x}(i)y_r(i)]$ and the interference-plus-noise correlation vector $E[\mathbf{n}(i)y_{nr}(i)]$, respectively. The *adaptation sequence* is defined as $\alpha(k) = \mu_0(k)[\widehat{\rho}(k) + 1]$. Substituting these estimates and the preceding results into equation 15, we obtain the *maximin algorithm*:

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \alpha(k)\left[\frac{\mathbf{c}_x(k)}{\hat{p}_x(k)} - \frac{\mathbf{c}_n(k)}{\hat{p}_n(k)}\right], \qquad k \geq 0 \tag{22}$$

where $\mathbf{w}(0)$ is the deterministic initial weight vector. As the adaptive weights converge, the interference components of $\mathbf{c}_x(k)$ and $\hat{p}_x(k)$ decrease. Thus, the first term within the brackets can be interpreted as a *signal term* that enables the algorithm to direct the array beam toward the desired signal. The second term within the brackets is a *noise term* that enables the algorithm to null interference signals.

The adaptation sequence $\alpha(k)$ should be chosen so that $E[\mathbf{w}(k)]$ converges to a nearly optimal steady-state value. It is also intuitively plausible that $\alpha(k)$ should decrease rapidly as $E[\mathbf{w}(k)]$ converges. A suitable candidate is

$$\alpha(k) = \alpha \frac{\hat{p}_n(k)}{\hat{t}(k)} \tag{23}$$

where $\hat{t}(k)$ is an estimate of the total interference and noise power entering the array, and α is the *adaptation constant*. The convergence analysis of appendix B and the subsequent simulation results confirm that this choice is effective and robust, provided that $0 < \alpha < 4$. Another adaptation sequence that works well (*2*), (*3*) is $\alpha(k) = \alpha\hat{p}_n(k)/\hat{p}_x(k)$, but the best choice for α in this sequence depends on the SINR at each input to the adaptive filter.

The remaining issue is the choice of estimators for $\hat{t}(k), \mathbf{c}_x(k), \mathbf{c}_n(k), \hat{p}_x(k)$, and $\hat{p}_n(k)$. The specific nature of the spread-spectrum signals allows blind estimates to be made without depending on known steering vectors or reference signals.

3. Maximin Processor

To implement the maximin processor, it is necessary to separate the interference from the total signal $\mathbf{x}(i)$, which includes both the desired signal and the interference. After each hop, the frequency-hopping signal has a carrier frequency f_h and its spectrum is largely confined to a frequency channel with one-sided bandwidth B, as depicted in figure 2. In each branch of the maximin processor following an array antenna, the frequency hopping is removed and the current frequency channel or *signal channel* is downconverted to baseband. A *signal filter* then extracts the total signal in the signal channel. To cancel the interference imbedded in $\mathbf{x}(i)$, the receiver measures the interference in a *monitor channel*, which is a nearby frequency channel or spectral region with a center frequency offset by

$$f_o \geq B \tag{24}$$

from the carrier frequency. For this measurement, the maximin processor in each branch downconverts the monitor channel to baseband. After the downconversion, a baseband

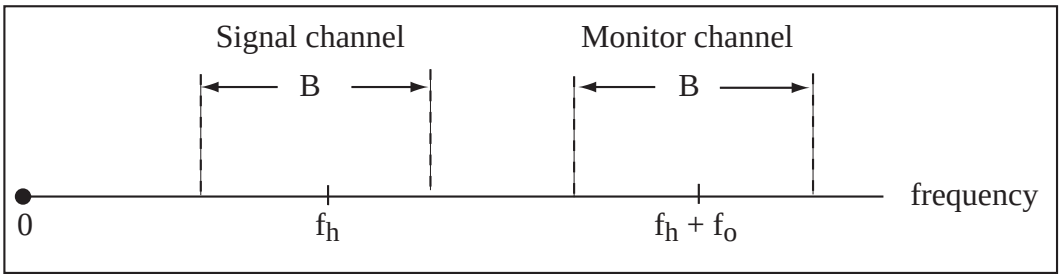

Figure 2. Signal channel and monitor channel during hop dwell time.

monitor filter extracts the interference in the monitor channel. Ideally, the spectrum of an interference signal overlaps the signal and monitor channels so that the interference components in the signal-filter and monitor-filter outputs have the same second-order statistics. After a sufficient number of hops, each monitor filter will process most of the interference that originates from a particular direction and spectrally overlaps the hopping band. The outputs of all the branch signal and monitor filters are used by the maximin algorithm to enable interference cancellation and desired-signal enhancement. If spectral splatter of the desired-signal spectrum into the monitor filter may lead to significant desired-signal cancellation by the adaptive algorithm, then $f_o > B$ may be necessary.

Figure 3 illustrates the principal components of the initial processor behind an antenna of an adaptive array for a frequency-hopping system. The desired frequency-hopping signal that arrives at array antenna k is

$$s_k(t) = \cos[2\pi f_h(t)t + \phi(t - t_{k0}) + \phi_{k0}] \tag{25}$$

where $f_h(t)$ represents the time-varying carrier frequency, $\phi(t)$ represents the data modulation, t_{k0} is the delay at antenna k, and ϕ_{k0} is the phase shift at antenna k. The front-end devices include a bandpass filter that excludes noise outside the hopping band. Dehopping by mixing the received signal with a local synchronized frequency-hopping replica changes the variable carrier frequency into a fixed intermediate frequency (IF) denoted by f_i. In the implementation shown in figure 3, the IF signal passes through an antialiasing IF filter and then is sampled by an analog-to-digital (A/D) converter. Let T_s denote the symbol duration. For some positive integer L, the sampling interval is $T_0 = T_s/L$. It is assumed that the antennas are close enough that the different delays at the antennas cause a negligible variation in the data modulation, and thus $t_{k0} = t_0$. Assuming perfect timing synchronization of the local frequency-hopping replica, the discrete-time IF desired signal is

$$s_k(i) = \cos[2\pi(f_i + \delta)T_0 i + \phi(T_0 i - t_0) + \phi_{k0}] \tag{26}$$

where δ denotes the carrier-offset error resulting from imperfect frequency synchronization of the local frequency-hopping replica.

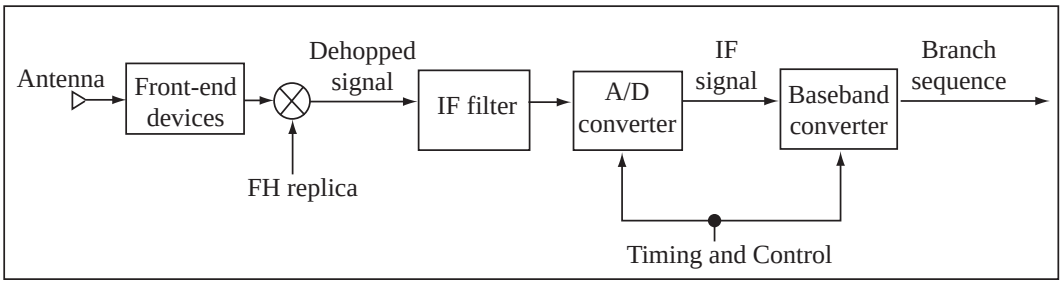

Figure 3. Dehopping and initial processing in a branch.

A complex-valued branch sequence is produced when the sampled signal is digitally downconverted to baseband and its complex envelope is extracted. The baseband converter first produces the in-phase and quadrature components of $s_k(i)$. The quadrature component is produced by passing $s_k(i)$ through a digital Hilbert transformer (*10*) while the in-phase component is produced by delaying $s_k(i)$ appropriately. The in-phase component is the real part and the quadrature component is the imaginary part of the complex-valued analytic signal, which has no positive frequency components. The spectrum of the analytic signal is downconverted to baseband by multiplying the analytic signal by $exp(-j2\pi f_i T_0 i)$, which gives the desired component of the kth branch sequence:

$$s_{bk}(i) = \exp\left\{j[2\pi\delta T_0 i + \phi(T_0 i - t_0) + \phi_{k0}]\right\}. \tag{27}$$

This component is the discrete-time complex envelope of the dehopped desired signal. The phase shift ϕ_{k0} reflects the different arrival times of the desired signal at the array antennas. Instead of sampling the IF signal, a direct downconversion of the IF signal to baseband is possible but entails the use of four mixers and their associated filters in each branch to accommodate the production of in-phase and quadrature components for both the signal and monitor channels.

To capture the energy in both the signal and monitor channels, the IF filter passband must encompass the *observed band* $f_i - B/2 \le |f| \le f_i + f_o + B/2$, where B is large enough to accommodate the carrier-offset error δ. To prevent aliasing, the upper cutoff frequency of the filter must be no larger than $1/2T_0 = L/2T_s$, which is half the sampling rate. Therefore, the number of samples per symbol must satisfy

$$L \ge 2T_s(f_i + f_o + B/2). \tag{28}$$

To prevent the overlapping of the positive and negative frequency components in the observed band, which will distort the spectrum of the filter output, it is necessary that

$$f_i \ge B/2. \tag{29}$$

The downconversion of the analytic signal can be simplified (*10*) if $f_i = 1/4T_0 = L/4T_s$. Substitution into the previous inequalities indicates that both are satisfied if

$$L \geq 4T_s(f_o + B/2), \qquad f_i = L/4T_s. \tag{30}$$

In the maximin processor illustrated in figure 4, each branch sequence is applied to a signal filter. A further downconversion of each branch sequence by f_o provides the complex envelopes of the monitor-channel signals. The branch sequence $\mathbf{x}_1(i)$, each component of which is a signal-filter input, is used to produce

$$\mathbf{x}_2(i) = \mathbf{x}_1(i) \exp(-j2\pi f_o T_0 i) \tag{31}$$

each component of which is a monitor-filter input. The baseband signal and monitor filters are identical with passbands such that $|f| \leq B/2$ so that the interference components of the two filter outputs are similar. The signal-filter and the monitor-filter outputs are components of $\mathbf{x}(i)$ and $\hat{\mathbf{n}}(i)$, respectively, the vectors applied to the adaptive filter. The vector $\hat{\mathbf{n}}(i)$ provides an estimate of the interference and noise in $\mathbf{x}(i)$.

The adaptive filter executes the maximin algorithm, which seeks to maximize the SINR at the input to the digital demodulator. The architecture of the adaptive filter is illustrated in figure 5. The vectors applied to the adaptive filter are $\mathbf{x}(i)$ and $\hat{\mathbf{n}}(i)$, and m samples are generated per weight iteration. The adaptive filter produces the output

$$y_r(i) = Re[\mathbf{w}^H(k)\mathbf{x}(i)], \quad i = km + 1, \cdots, (k+1)m \tag{32}$$

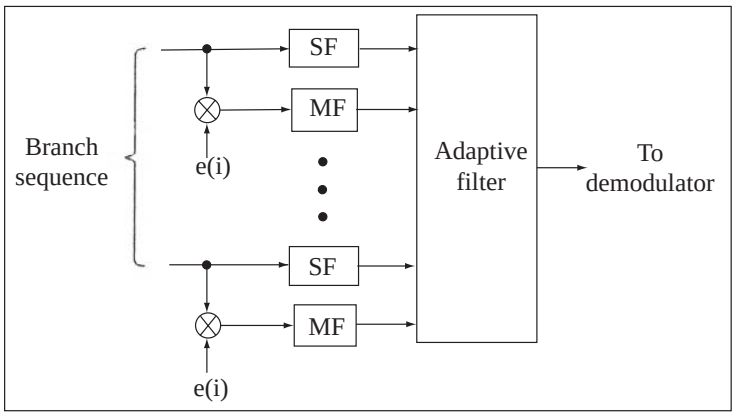

Figure 4. Maximin processor with SF = signal filter, MF = monitor filter, and e(i) = $exp(-j2\pi f_o i T_s/L)$.

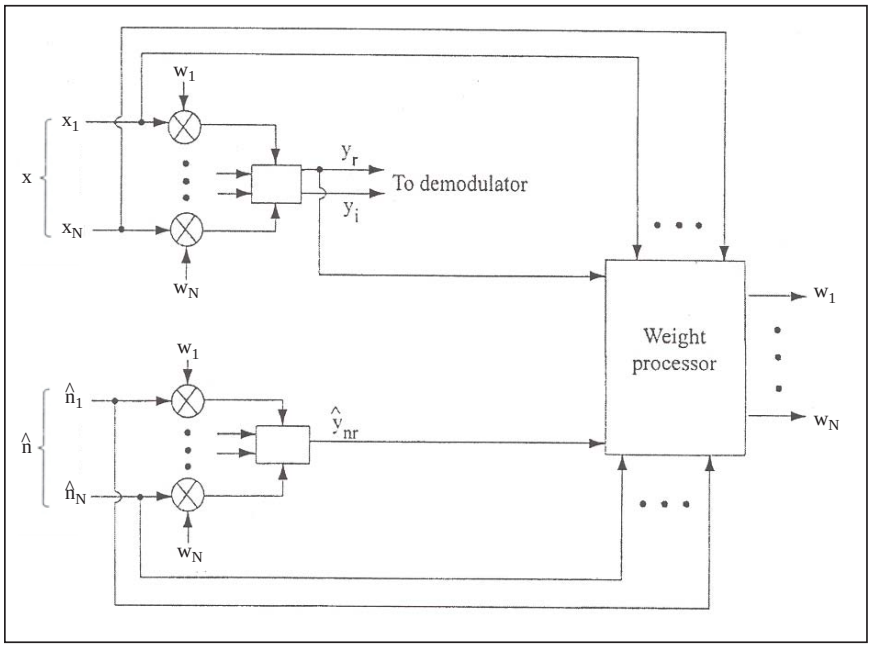

Figure 5. Adaptive filter that executes maximin algorithm.

where sample i is taken after weight iteration k. This output is applied to the demodulator and is used in the estimators

$$\mathbf{c}_x(k) = \frac{1}{m} \sum_{i=km+1}^{(k+1)m} \mathbf{x}(i) y_r(i), \qquad k \geq 0 \tag{33}$$

and

$$\hat{p}_x(k) = \frac{1}{m} \sum_{i=km+1}^{(k+1)m} y_r^2(i), \qquad k \geq 0 \tag{34}$$

which are unbiased when $\mathbf{x}(i)$ and $y_r(i)$ are stationary processes between weight iterations.

The adaptive filter also generates

$$\hat{y}_{nr}(i) = Re[\mathbf{w}^H(k)\hat{\mathbf{n}}(i)], \quad i = km+1, \cdots, (k+1)m \ . \tag{35}$$

The critical requirement of $\hat{\mathbf{n}}(i)$ is for it to have the same second-order statistics as $\mathbf{n}(i)$, the undesired component of $\mathbf{x}(i)$. If the spectrum of an interference signal overlaps the signal and monitor channels, then the vector

$$\mathbf{n}_e(i) = \hat{\mathbf{n}}(i) \exp(-j2\pi f_o i T_s) \tag{36}$$

provides an estimate of the signal-filter output $\mathbf{n}(i)$. Treating the weight vector as a constant, observing that $E[\mathbf{n}(i)\mathbf{n}^H(i)] \approx E[\mathbf{n}_e(i)\mathbf{n}_e^H(i)] = E[\hat{\mathbf{n}}(i)\hat{\mathbf{n}}^H(i)]$, and calculating as in equation 18, we obtain

$$E[\hat{\mathbf{n}}(i)\hat{y}_{nr}(i)] \approx E[\mathbf{n}(i)y_{nr}(i)]. \tag{37}$$

Therefore, an estimator of the interference-plus-noise correlation vector at weight iteration k is

$$\mathbf{c}_n(k) = \frac{1}{m}\sum_{i=km+1}^{(k+1)m} \hat{\mathbf{n}}(i)\hat{y}_{nr}(i), \qquad k \geq 0. \tag{38}$$

Similarly, an estimator proportional to the interference-plus-noise output power is

$$\hat{p}_n(k) = \frac{1}{m}\sum_{i=km+1}^{(k+1)m} \hat{y}_{nr}^2(i), \qquad k \geq 0. \tag{39}$$

Both of these estimators are unbiased when $\hat{\mathbf{n}}(i)$ and $\hat{y}_{nr}(i)$ are stationary processes between weight iterations.

A recursive estimator of the total interference-plus-noise power entering the array is

$$\hat{t}(k) = \begin{cases} \mu\hat{t}(k-1) + \frac{1-\mu}{m}\sum_{i=km+1}^{(k+1)m} \parallel \hat{\mathbf{n}}(i) \parallel^2, & k \geq 1 \\ \frac{1}{m}\sum_{i=1}^{m} \parallel \hat{\mathbf{n}}(i) \parallel^2, & k = 0 \end{cases} \tag{40}$$

where $\parallel \mathbf{v} \parallel$ denotes the Euclidean norm of $\mathbf{v}$, μ is the *memory factor*, and $0 \leq \mu \leq 1$. This estimator is unbiased when $\hat{\mathbf{n}}(i)$ is a wide-sense stationary process. The memory factor is useful in a nonstationary environment, which may be due to the presence of partial-band interference. If $\mu = 0$, then when the desired signal occasionally hops into a frequency channel with interference, the adaptation sequence may at first be too small to enable the rapid adaptation of the maximin algorithm to this interference. However, if μ is sufficiently large, then the maximin algorithm responds rapidly to partial-band interference.

The output sequence produced by the adaptive filter is applied to the digital demodulator. Ideally, the desired-signal component of the demodulator input should be the undistorted discrete-time complex envelope of the desired signal. Toward this goal, the signal filters in the maximin processor should be designed to introduce negligible distortion over the demodulator passband. If such filters are not feasible, then the demodulator input can be obtained by the processing shown in figure 6. The weight vector computed by the adaptive filter is applied to a device that computes the inner product of this vector and $\mathbf{x}(i)$, the vector of branch sequences, each of which includes an undistorted discrete-time desired sequence. The resulting inner product is applied to the demodulator, which provides the appropriate filtering.

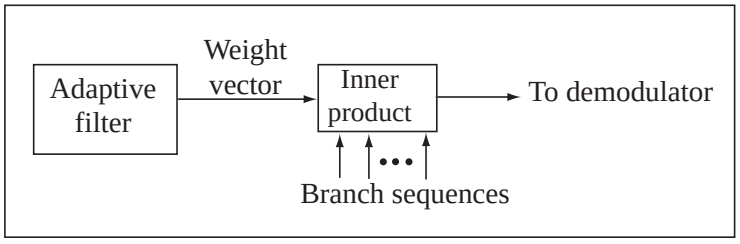

Figure 6. Processing for no distortion of demodulator input.

If the circular symmetry is not used in the derivation of the maximin algorithm, then an analogous derivation yields a slightly more complicated alternative algorithm (appendix C). Simulation experiments indicate that the alternative algorithm offers no significant performance advantage relative to the maximin algorithm, despite requiring twice as much computation, and tends to generate very large weights unless the weight vector is normalized. Thus, the alternative algorithm is not considered further.

Multipath components of the desired signal are not cancelled by the maximin algorithm because their spectra occupy only the signal channel and are insignificant in the monitor channel. However, the beamforming generated by the algorithm often excludes those multipath signals that arrive from directions much different from the direction of the main frequency-hopping signal.

Prior to acquisition of the frequency-hopping pattern (*6*), the frequency synthesized by the receiver differs from the received carrier frequency. Consequently, the desired signal is usually absent from the signal-filter outputs and the maximin algorithm is ineffective. The power-minimization algorithm (*11*) might be used to assist in the acquisition of the frequency-hopping pattern by forming nulls in the directions of the interference signals. However, it will rarely be necessary because frequency hopping inherently produces time intervals with little or no interference in the signal channel. During these intervals, a single antenna output can be processed to enable rapid acquisition. After verification of the acquisition, the full adaptive array and the maximin algorithm may be activated.

The computational cost per weight iteration of the maximin algorithm can be estimated in terms of the number of real multiplications, real additions, and real divisions.

1. Equation 32 indicates that $y_r(i)$ requires the evaluation of an inner product with N terms. Since each term requires the calculation of only the real part of a complex multiplication, $y_r(i)$ requires $2N$ real multiplications. Similarly, $\hat{y}_{nr}(i)$ requires $2N$ real multiplications.

2. Since the factor $1/m$ cancels in the ratios appearing in the maximin algorithm, it suffices to calculate $m\hat{p}_x(k)$, $m\hat{p}_n(k)$, $m\hat{t}(k)$, $m\mathbf{c}_x(k)$, and $m\mathbf{c}_n(k)$. Equation 34 indicates that $m\hat{p}_x(k)$ requires m real multiplications for the squaring operations and $m-1$ real additions. Similarly, $m\hat{p}_n(k)$ requires m real multiplications and $m-1$ real additions. Each squared norm in equation 40 requires $2N$ real squaring operations and $2N-1$ real additions. Thus, $m\hat{t}(k)$ requires $2Nm+3$ real multiplications and $(2N-1)m+1$ real additions. Equations 33 and 38 indicate that each component of $m\mathbf{c}_x(k)$ or $m\mathbf{c}_n(k)$ requires m multiplications of a complex number by a real number, which requires $2m$ real multiplications. Each component requires $m-1$ complex additions or $2(m-1)$ real additions. Thus, both $m\mathbf{c}_x(k)$ and $m\mathbf{c}_n(k)$ require $2Nm$ real multiplications and $2N(m-1)$ real additions.

3. Equation 23 indicates that the calculation of $\alpha(k)$ requires 1 real multiplication and 1 real division.

4. Once all the factors in equation 22 have been calculated, the evaluation of each component of equation 22 requires 4 real divisions, 2 real multiplications, and 2 complex additions or 4 real additions. Thus, the evaluation of equation 22 requires $4N$ real divisions, $2N$ real multiplications, and $4N$ real additions.

Summing the preceding results, it is found that each iteration of the maximin algorithm requires $4N+1$ real divisions, $6Nm+6N+2m+4$ real multiplications, and $6Nm+m-1$ real additions. Therefore, the computational cost of the maximin algorithm per iteration is $O(mN)$ real multiplications or divisions and $O(mN)$ real additions. This cost is of the same order as that of the partial-rank algorithm (*8*). The computational cost of the maximin algorithm per sampling interval is $O(N)$ real multiplications or divisions and $O(N)$ real additions, which is on the order of the cost of the classical least-mean-squares (LMS) algorithm. In the subsequent simulation experiments, $N=4$ and $m=100$, which implies that the cost of each weight iteration of the maximin algorithm is 17 real divisions, 2628 real multiplications, and 2499 real additions.

4. Anticipative Maximin Algorithm

The monitor filter produces an approximation or estimate of the interference and then uses this estimate for interference cancellation. The monitor filter produces this estimate by observing a spectral region that has little desired-signal energy. An *anticipative filter* can observe a temporal region without desired-signal energy to produce another interference estimate that potentially allows additional cancellation by the anticipative maximin algorithm. This capability ultimately is due to the receiver's knowledge of the frequency-hopping pattern.

As illustrated in figure 7, the dehopping and initial processing in each of two parallel subbranches is similar to that shown in figure 3. The received signal is mixed with sinusoids at frequencies $f_h - f_i$ and $f_{h1} - f_i$, where f_h is the current carrier frequency, f_{h1} is the carrier frequency after the next hop, and f_i is the intermediate frequency. The upper subbranch produces the same output as the branch of figure 3, while the lower subbranch produces an output derived from the next signal channel. Each IF filter 1 has a passband such that $f_i - B/2 \leq |f| \leq f_i + f_o + B/2$, which corresponds to the current signal and monitor channels. Each IF filter 2 has a passband that covers $f_i - B/2 \leq |f| \leq f_i + B/2$, which corresponds to the next signal channel. The parameters f_i and L satisfy equations 30 or 28 and 29. The parallel A/D and baseband converters are identical and produce the digital samples of the main branch sequence and the anticipative branch sequence, respectively.

These sequences are applied to the anticipative maximin processor of figure 8. The main processor, which is identical to the maximin processor of figure 4, produces the demodulator input. Each anticipative branch sequence is applied to a monitor filter with a passband such that $|f| \leq B/2$. The output of each monitor filter is a component of the vector $\hat{\mathbf{n}}_a(i)$, which estimates the interference-plus-noise that will be present in the signal channel after the next frequency hop. The signal-filter outputs of the main processor provide the same vector $\mathbf{x}(i)$ used by the adaptive filter in the main processor. Both $\hat{\mathbf{n}}_a(i)$ and $\mathbf{x}(i)$ are applied to the anticipative adaptive filter. The anticipative filter adapts its weight vector, which converges toward the optimal weight vector for the next carrier frequency. After each hop, the weight vector associated with the new carrier frequency is transferred from the anticipative filter to the main processor. Transfers are triggered by the clock that controls the frequency-hopping carrier transitions.

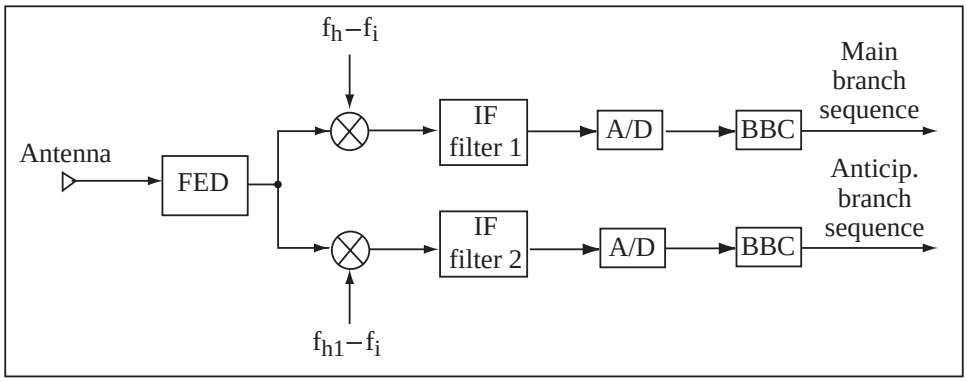

Figure 7. Dehopping and initial processing in two parallel subbranches of a branch. FED=front-end devices; BBC=baseband converter.

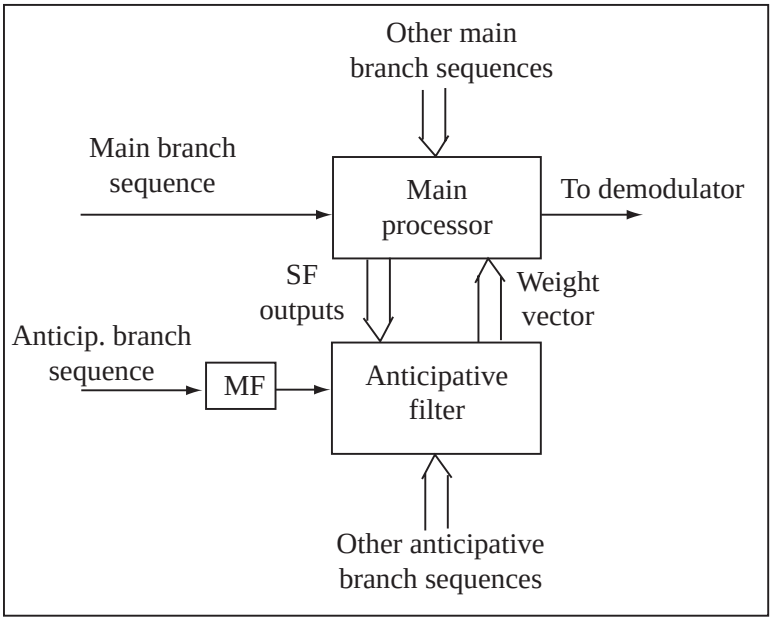

Figure 8. Anticipative maximin processor. MF=monitor filter.

The anticipative weight vector $\mathbf{w}_a(k)$ is computed by the anticipative filter at iteration k. The recursive equation is a modified version of the maximin algorithm:

$$\mathbf{w}_a(k+1) = \mathbf{w}_a(k) + \left\{ \alpha(k) \left[\frac{\mathbf{c}_x(k)}{\hat{p}_x(k)} - \frac{\mathbf{c}_n(k)}{\hat{p}_n(k)} \right] \right\}_a, \qquad k \geq 0 \tag{41}$$

where the subscript a denotes a quantity computed by the anticipative filter. The adaptation constant and the memory factor used in this equation are set equal to their values in the main processor. The weight vector in the main processor is updated by computing equation 22, except at sampling instants occurring at the end of a dwell interval or during switching times. At these instants, the weight vector in the main processor is set equal to that of the anticipative filter. The switching times occur when $k = nk_0$ in the main processor, where k_0 is the number of iterations per hop and $n = 1, 2, \ldots$ is the hop number. Thus, the algorithm in the main processor becomes

$$\begin{aligned} \mathbf{w}(k+1) &= \mathbf{w}(k) + \alpha(k) \left[\frac{\mathbf{c}_x(k)}{\hat{p}_x(k)} - \frac{\mathbf{c}_n(k)}{\hat{p}_n(k)} \right], \qquad k+1 \neq nk_0, \quad k \geq 0 \\ \mathbf{w}\left(nk_0\right) &= \mathbf{w}_a\left(nk_0\right), \qquad n = 0, 1, \ldots \end{aligned} \tag{42}$$

Equations 41 and 42 constitute the *anticipative maximin algorithm.*

The computational cost per iteration of the anticipative maximin algorithm is nearly double that of the maximin algorithm if $k_0 >> 1$. From the previously evaluated computational cost of the maximin algorithm, the computational cost per iteration of the

anticipative maximin algorithm is approximately $8N + 2$ real divisions, $12Nm + 12N + 4m + 8$ real multiplications, and $12Nm + 2m - 2$ real additions. In the subsequent simulation experiments, $N = 4$ and $m = 100$, which implies that the cost of each weight iteration of the anticipative maximin algorithm is approximately 34 real divisions, 5256 real multiplications, and 4998 real additions.

The anticipative maximin algorithm expedites the convergence of the mean weight vector in the presence of stationary partial-band interference at the cost of additional hardware and nearly a doubling of the computational requirements of the maximin algorithm.

5. Simulation Experiments

In the simulation experiments, the array consists of 4 omnidirectional antennas located at the vertices of a square. Let λ denote the wavelength corresponding to the center frequency of the desired signal, which is 3 GHz. The edge length or the separation between adjacent antennas is $d = \lambda$ or $d = 1.5\lambda$. All signals are assumed to arrive as plane waves with no fading. The frequency-hopping signal is modulated by binary minimum-shift keying (MSK) and has a carrier frequency that is randomly chosen from a hopset. Each frequency channel, which includes a hopset frequency, has a bandwidth $B = 100$ kHz. The hopping band has a bandwidth W_h, and there are W_h/B contiguous frequency channels. The hop dwell time is 1 ms. The frequency-hopping signal arrives from a direction 20 degrees counterclockwise from the normal to one of the edges and has a frequency offset equal to $\delta = 1$ kHz after downconversion, which models imperfect frequency synchronization. Perfect timing synchronization of the local frequency-hopping replica is assumed. The sequence of data bits is randomly generated at the rate of $1/T_s = 100$ kbps. The sample rate is 1.0 megasamples per second, which corresponds to $L = 10$ samples per bit. The signal and monitor filters are modeled as digital Chebyshev filters of the second kind (*10*) with the 3-dB bandwidths equal to B. The Chebyshev filter was selected because it has a mild group-delay variation with frequency that tends to limit signal distortion. The monitor channel is offset by the minimal amount $f_o = 100$ kHz because the effects of contamination by the desired signal were found to be minor when the Chebyshev filters are used. Thus, equation 30 is satisfied if $f_i = 250$ kHz. The thermal noise in each branch sequence is modeled as bandlimited complex Gaussian noise. The anticipative maximin and maximin algorithms are implemented with $\alpha = 0.1$ and $\mu = 0.99$. These values usually provide close to the best overall performance against the modeled partial-band interference. A weight iteration occurs after each 10 data bits. The weight-iteration rate and the adaptation constant are both partly selected to ensure that the weights do not increase to excessively large values. For each simulation trial, the initial weight vector of each adaptive processor is $\mathbf{w}(0) = [1 \;\; 0 \;\; 0 \;\; 0]$, which forms an omnidirectional array pattern. The IF filters are modeled as ideal rectangular filters. The signal-to-noise ratio (SNR) is 14 dB at the output

of each signal filter and monitor filter in the adaptive processors. Table 1 lists the principal system parameter values used in the simulation experiments unless otherwise stated.

High-power, partial-band interference is a realistic and significant threat to frequency-hopping communications. Since partial-band interference is not always present in the monitor channel, it cannot be cancelled as rapidly as full-band interference. Interference that occupies only a small part of the hopping band, or even frequency-hopping interference signals, can be suppressed by the adaptive array supplemented by an error-control code. If the interference is observed by a monitor filter often enough, it is rapidly suppressed by the adaptive array; if it is observed only occasionally, then the interference is primarily suppressed by the error-control code. In the simulation experiments, each of 1, 2, or 3 interference signals has its power distributed among equal-power tones (continuous-wave signals) in frequency channels that cover the fraction 0.1 of the hopping band unless otherwise stated. For each interference signal, the interference-to-signal ratio (ISR) is equal to 0 dB in each frequency channel that contains a tone. After the downconversions, the interference signals have different initial phase shifts and residual frequency offsets equal to 10 kHz, 13 kHz, and 16 kHz, respectively, which reflect the mismatch of the tone frequencies and the hopset carrier frequencies. Multiple interference signals do not add coherently at all antennas, even if they have the same carrier frequencies, because they arrive from different directions and have different initial phase shifts. Unless otherwise stated, it is assumed that interference occupies contiguous

Table 1. Basic system parameters.

Parameter	**Value**
Array geometry	square with 4 antennas
Center frequency	3 GHz
Antenna separation	λ or 1.5λ
Direction of desired signal	20°
Hop dwell time	1 ms
Modulation	MSK
Data rate	100 kbs
Filter bandwidths	100 kHz
Monitor channel offset	100 kHz
Frequency offset of desired signal	1 kHz
Sampling rate	1.0 Ms/s
SNR at each SF or MF output	14 dB
Adaptation constant	$\alpha = 0.1$
Memory factor	$\mu = 0.99$
ISR per channel for each interf. signal	0 dB
Fraction of band covered by interf. sig.	0.1
Frequency offsets of interference	10, 13, and 16 kHz
Initial weight vector	[1 0 0 0]
Bits per weight iteration	10

frequency channels and that an interference tone in the signal filter is always accompanied by a tone in the monitor filter. The latter assumption is a good approximation for partial-band interference over a substantial fraction of the hopping band when the signal and monitor channels are adjacent. The SINR at the processor output is calculated after each sample time and then averaged over all samples in the time interval between a weight iteration and a preceding one to determine the SINR at each weight iteration. The SINR is observed to fluctuate, but tends to gradually increase until it reaches a steady-state condition with a smaller residual fluctuation.

Figures 9 to 17 show the results for typical simulation trials in which $d = \lambda$, $W_h = 30$ MHz, and there are 100 frequency hops per trial. A typical trial for the maximin algorithm, the parameter values of table 1, and one interference signal arriving at a 40° angle with a frequency offset equal to 10 kHz is illustrated in figure 9, which shows the SINR variation with the weight iteration number. The SINR does not drop below 0 dB the first time the interference is encountered in the signal channel because beamforming in the desired-signal direction has already occurred and a partial grating null near the interference-signal direction has already been formed. Between observations of the interference signal, which is observed only 10 percent of the time, the weights tend to slowly drift toward the values they would have without the interference. A steady-state condition is gradually reached, and subsequent SINR fluctuations tend to be comparable to those that would occur in the absence of interference. Since the arrival angle of the interference is only 20° apart from the arrival angle of the desired signal, the angular separation between the two signals is within the half-power beamwidth of the array. Much better performance against a single interference signal is achieved if the angular separation exceeds the half-power beamwidth. In figure 10, the conditions are identical to those of figure 9 except that the anticipative maximin algorithm is used. A faster convergence to the steady state is observed.

Let θ denote an arrival angle defined as the angle in the counterclockwise direction from the normal to one of the array edges. Let $\mathbf{s}_r(\theta)$ denote the *steering vector* or *array response vector*, which is the array response to an ideal plane wave arriving at angle θ (*6*), (*7*), (*8*). For a square array with 4 antennas, the components of the steering vector are

$$\begin{aligned} s_{r1} &= 1, \quad s_{r2} = \exp(-j2\pi\frac{d}{\lambda}\sin\theta) \\ s_{r3} &= \exp(-j2\pi\frac{d}{\lambda}\cos\theta), \quad s_{r4} = \exp[-j2\pi\frac{d}{\lambda}(sin\theta + \cos\theta)]. \end{aligned} \tag{43}$$

The array gain pattern after weight iteration k is

$$G(\theta, k) = \frac{\left|\mathbf{w}^H(k)\mathbf{s}_r(\theta)\right|^2}{\left\|\mathbf{w}(k)\right\|^2}. \tag{44}$$

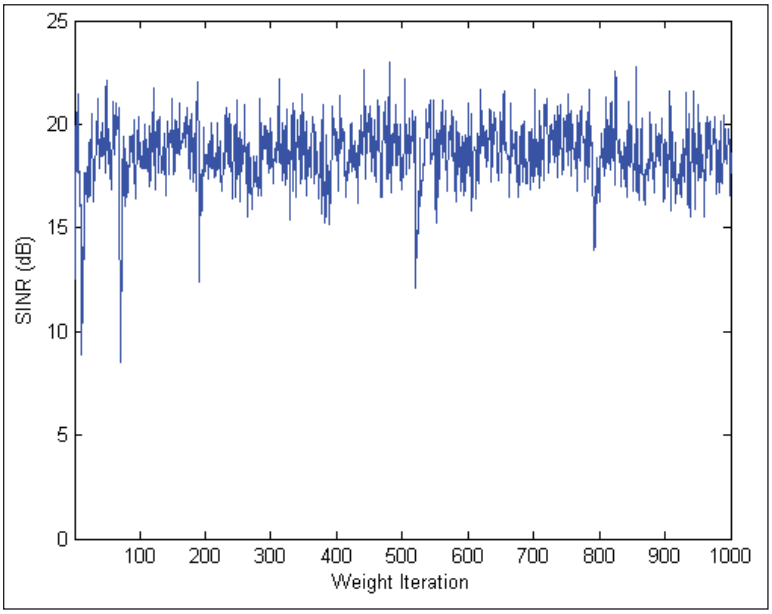

Figure 9. SINR in simulation trial for maximin algorithm with $W_h = 30$ MHz and one interference signal at 40°.

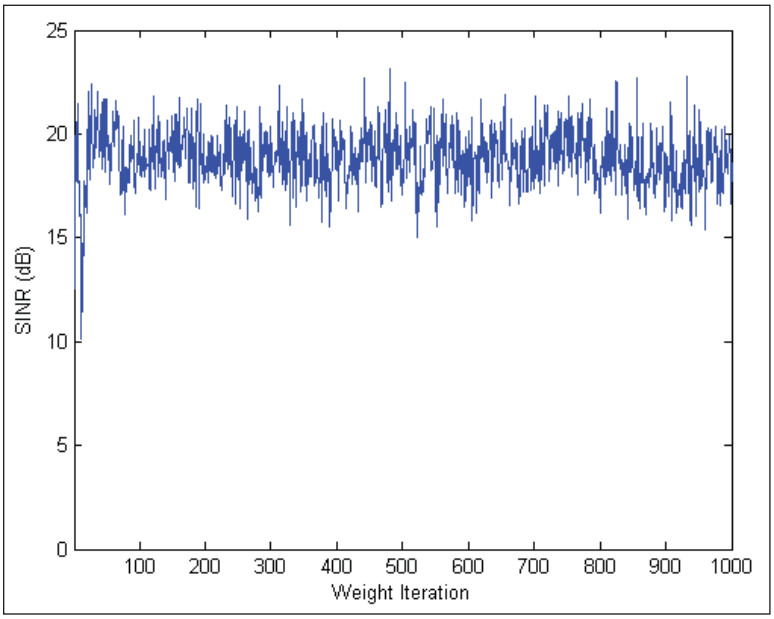

Figure 10. SINR in simulation trial for anticipative maximin algorithm with $W_h = 30$ MHz and one interference signal at 40°.

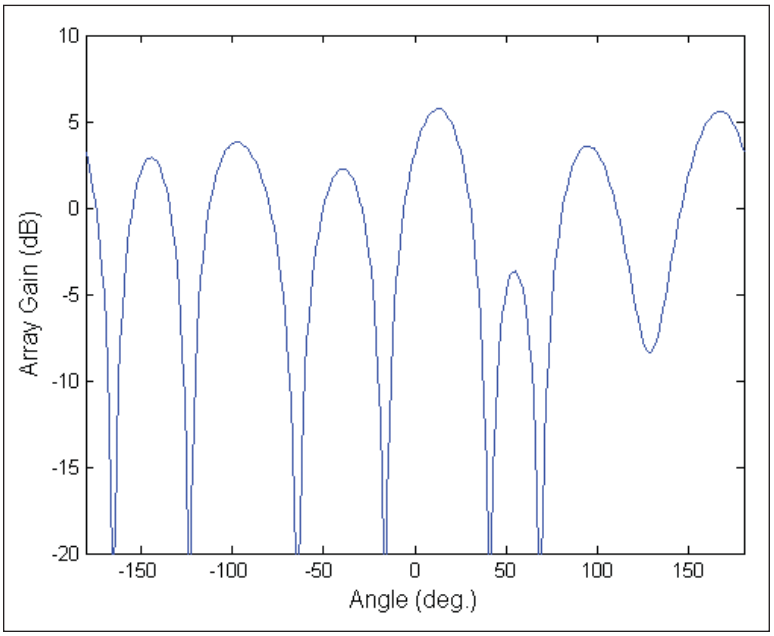

Figure 11. Array gain pattern at end of simulation trial for anticipative maximin algorithm with $W_h = 30$ MHz and one interference signal at 40°.

The array gain pattern at the end of the simulation trial of figure 10, which is almost identical to the one at the end of the simulation trial of figure 9, is depicted in figure 11. A null deeper than −20 dB in the approximate direction of the interference signal and a mainlobe slightly displaced from the direction of the desired signal have formed along with other grating nulls and grating lobes.

Figure 12 depicts the SINR variation in a typical simulation trial under the conditions of figure 9 except that $\mu = 0$. The absence of convergence to the steady-state performance of figure 9 after 100 hops is apparent. The reason is the sporadic large decrease in the magnitude of the adaptation sequence, which prevents rapid weight adaptation when interference is present in the signal channel. In general, setting $\mu > 0$ is useful against partial-band interference and neutral in the absence of interference.

When more than one interference signal is present, the adaptation becomes more difficult as the array encounters a more rapidly varying signal environment. The performance of the adaptive algorithm is highly dependent on the angular separations among the arriving interference signals and the desired signal. Assuming that the interference signals are adequately separated from the desired signal, separations of the interference signals from each other on the order of a beamwidth can be much more easily accommodated than separations on the order of 0.5 or 1.5 beamwidths. Figure 13 illustrates the SINR variation in a typical simulation trial for the same conditions as figure 9, except that two interference signals arrive at angles 40° and −10° with frequency offsets of 10 and 13 kHz, respectively.

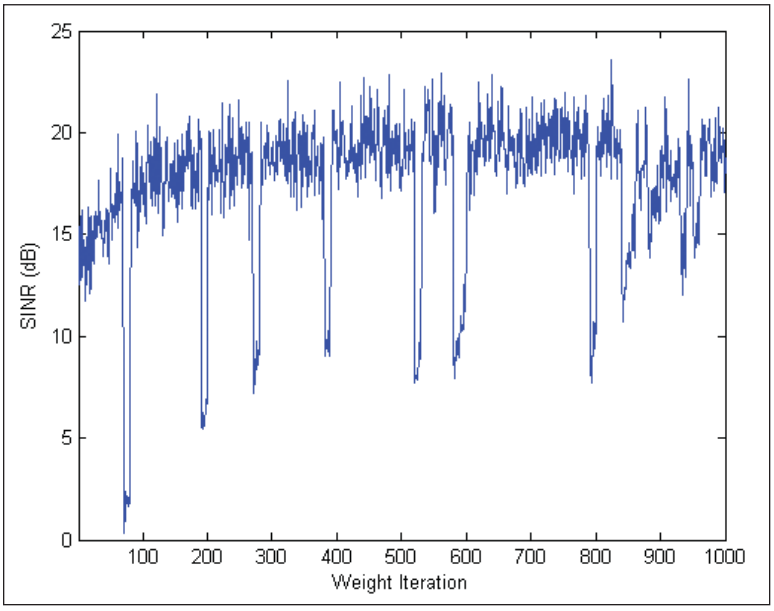

Figure 12. SINR in simulation trial for maximin algorithm with $W_h = 30$ MHz, one interference signal at 40°, and no memory factor.

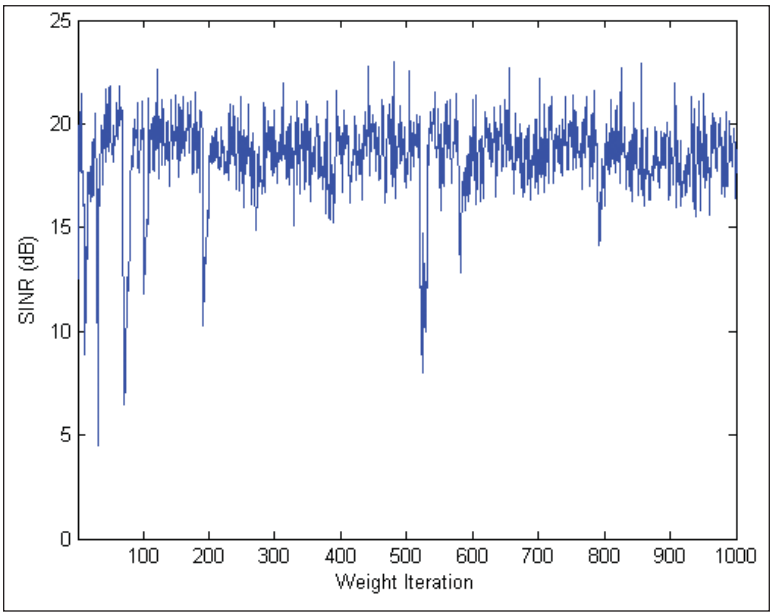

Figure 13. SINR in simulation trial for maximin algorithm with $W_h = 30$ MHz and two interference signals at 40° and −10°.

A comparison between figures 13 and 9 indicates that the convergence to steady state is slowed due to the presence of the second interference signal, but not greatly because the signal at −10° arrives at an angle near a grating null that naturally tends to form when the signal at 40° is nulled. As shown in figure 14, the anticipative maximin algorithm expedites convergence.

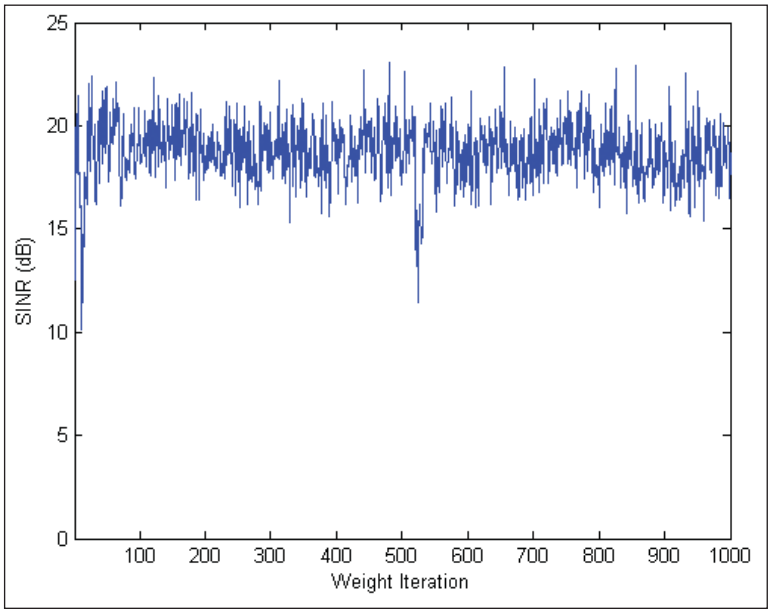

Figure 14. SINR in simulation trial for anticipative maximin algorithm with $W_h = 30$ MHz and two interference signals at $40°$ and $-10°$.

For the conditions of figure 13, if each interference signal occupies the fraction 0.5 of the hopping band, then the convergence is more rapid, as shown in figure 15. The increase in the convergence rate occurs because the monitor filter observes the interference for longer time intervals.

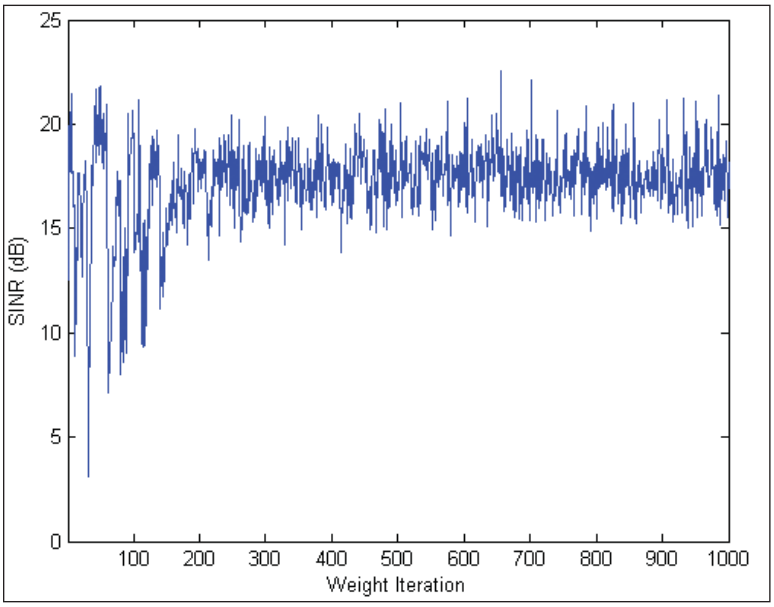

Figure 15. SINR in simulation trial for maximin algorithm with $W_h = 30$ MHz and two interference signals at $40°$ and -10°, each of which occupies 0.5 of the hopping band.

Figure 16 depicts the SINR variation in a typical simulation trial under the conditions of figure 13 except that the tones associated with each interference signal are randomly distributed throughout the hopping band. This scenario models sophisticated jamming. Since an interference tone in the signal channel does not necessarily imply a simultaneous interference tone in the monitor channel, the interference cancellation is impeded, and the convergence to a steady state is slowed. The anticipative maximin algorithm greatly improves the convergence rate, as illustrated in figure 17.

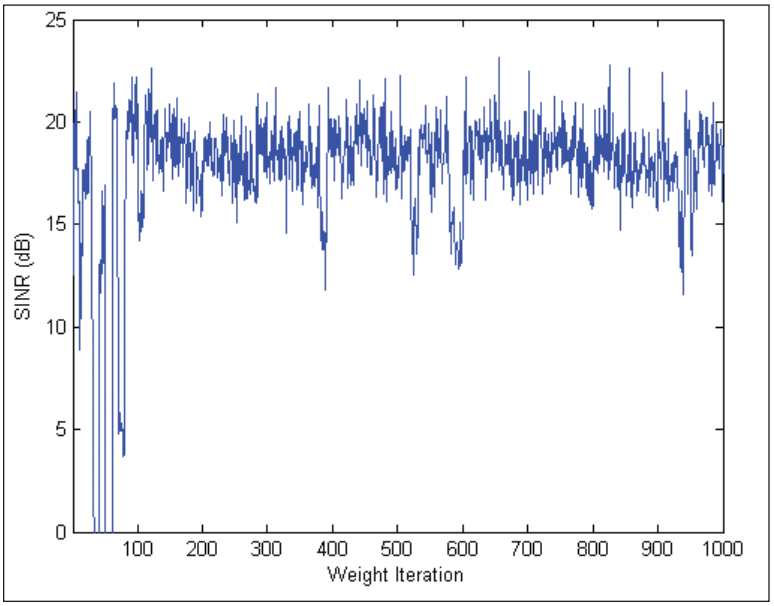

Figure 16. SINR in simulation trial for maximin algorithm with $W_h = 30$ MHz, two interference signals at 40° and −10, and randomly distributed tones.

The results of 10 representative simulation experiments for a 30 MHz bandwidth with $d = \lambda$ are summarized in table 2. The first six rows of table 2 show the statistics for the conditions of figures 9, 10, 13, 14, 16, and 17, respectively. The first column gives the arrival angles of 1, 2, or 3 interference signals. The second column indicates whether the anticipative maximin or the maximin algorithm is used. The third column indicates when the interference tones occupy randomly distributed rather than contiguous frequency channels. The SINRs for 20 simulation trials with 50 hops and 500 weight iterations per trial are averaged to obtain the *average SINR* and are used to calculate the standard deviation of the average SINR, which are listed in the fourth and fifth columns of the table. The standard deviation of the average SINR is an indicator of the rapidity and degree of convergence. The SINRs for the last 20 weight iterations of each of 20 simulation trials with 100 hops and 1000 weight iterations per trial are averaged to obtain the *final SINR* and are used to calculate the standard deviation of the final SINR, which are listed in the sixth and seventh columns of the table. The standard deviation of the final SINR is an indicator of the degree of convergence to a steady state after 80 hops. The final SINR is often less than the average SINR because the weight adaptation that enables interference

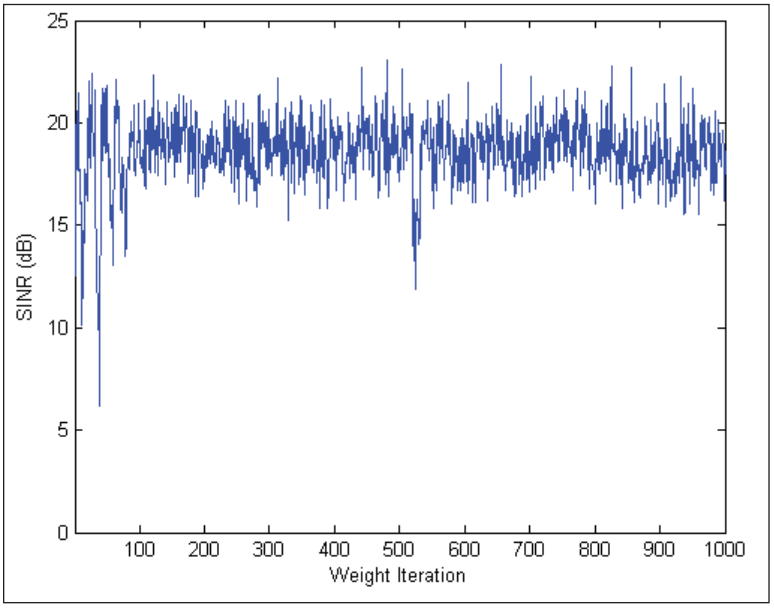

Figure 17. SINR in simulation trial for anticipative maximin algorithm with $W_h = 30$ MHz, two interference signals at 40° and −10°, and randomly distributed tones.

Table 2. Simulation results for $W_h = 30$ MHz and $d = \lambda$.

Interference Angles	**Anticip.**	**Rand. Distr.**	**Average SINR(dB)**	**Stand. dev.(dB)**	**Final SINR(dB)**	**Final St. dev.(dB)**
40°	no	no	18.67	1.73	18.81	1.41
40°	yes	no	19.06	1.48	19.04	1.35
40°, −10°	no	no	18.36	2.34	18.51	1.61
40°, −10°	yes	no	18.94	1.56	18.85	1.36
40°, −10°	no	yes	16.96	5.22	18.04	2.44
40°, −10°	yes	yes	18.70	2.61	18.78	1.40
40°, −10°, 85°	no	no	17.07	3.04	17.11	1.94
40°, −10°, 85°	yes	no	17.83	2.00	17.55	1.52
30°	no	no	16.06	3.19	15.91	2.31
30°	yes	no	16.96	2.44	16.72	2.12

cancellation reduces the SINR during interference-free dwell times. The decline in the average SINR as an interference signal is added, and hence the fraction of interfered frequency channels increases, is much less than the decline of nearly 2 dB that would occur without the adaptive algorithm. The array of 4 antennas is often able to substantially cancel 3 partial-band interference signals, as indicated in the table.

The final two rows illustrate the limitations imposed by the resolution of the array, which is its capability to distinguish between signals arriving at different angles. The statistics indicate the improvement provided by the anticipative maximin algorithm when the arrival

angles of the desired signal and the interference differ by only 10°, but the performance is still substantially degraded from the performance when the arrival angles differ by 20°. The resolution improves with increases in the array aperture or antenna separations. However, an enlarged aperture causes an increase in the number of grating lobes, which impedes the formation of nulls against two or more interference signals. Table 3 indicates the effects of an increase in the array aperture by increasing the edge length to $d = 1.5\ \lambda$. The superior resolution of the larger array yields greatly improved performance when a single interference signal arrives from nearly the same direction as the desired signal. The performance against two or three interference signals degrades, as is observed from a comparison of results listed in tables 2 and 3.

The effect of the ISR per channel is relatively mild. Table 4 lists the results of simulation experiments with 20 trials, 50 hops per trial, and one interference signal arriving at 40° with various power levels. It is observed that the convergence advantage of the anticipative maximin algorithm increases with the ISR per channel.

Table 3. Simulation results for $W_h = 30$ MHz and $d = 1.5\ \lambda$.

Interference Angles	**Anticip.**	**Rand. Distrib.**	**Average SINR(dB)**	**Stand. dev.(dB)**	**Final SINR(dB)**	**Final St. dev.(dB)**
40°	no	no	19.50	1.48	19.74	1.27
40°	yes	no	19.68	1.48	19.77	1.27
40°, −10°	no	no	17.27	2.83	17.17	2.05
40°, −10°	yes	no	17.92	1.92	17.67	1.62
40°, −10°	no	yes	15.98	4.93	16.88	2.64
40°, −10°	yes	yes	17.61	2.98	17.57	1.69
40°, −10°, 85°	no	no	15.68	3.67	15.32	2.32
40°, −10°, 85°	yes	no	16.52	2.31	15.92	1.67
30°	no	no	17.60	2.17	17.67	1.65
30°	yes	no	18.22	1.74	18.10	1.50

Table 4. Simulation results for $W_h = 30$ MHz, $d = \lambda$, and one interference signal at 40°.

ISR (dB)	**Anticipative**	**Average SINR (dB)**	**Standard dev. (dB)**
−10	no	19.12	1.64
−10	yes	19.45	1.42
0	no	18.67	1.73
0	yes	19.06	1.48
10	no	18.12	2.37
10	yes	18.69	1.74

6. Frequency Compensation

Adaptive processing may be impaired if the *fractional bandwidth*, which is defined as the hopping bandwidth divided by the center frequency of the hopping band, exceeds a few percent because the antennas produce only relative phase information. Consider two antennas of an adaptive array receiving a plane wave, as illustrated in figure 18. The phase shift of the signal component at frequency f that arrives at one element relative to the phase of the signal component at the other element is

$$\theta = \frac{2\pi d}{\lambda} \sin\phi = \frac{2\pi df}{c} \sin\phi \tag{45}$$

where c is the speed of an electromagnetic wave and ϕ is the arrival angle of the plane wave relative to a line perpendicular to the straight line through the two elements. If the frequency of the received signal component changes to $f + \Delta f$, then the phase shift changes by

$$\Delta\theta_1 = \frac{2\pi d\Delta f}{c} \sin\phi = \frac{2\pi d}{\lambda} \left(\frac{\Delta f}{f}\right) \sin\phi. \tag{46}$$

In most potential applications,

$$\left(\frac{\Delta f}{f}\right) \leq \frac{1}{2} \left(\frac{\lambda}{d}\right) \tag{47}$$

which implies that $|\Delta\theta_1| \leq \pi$ and that $\Delta\theta_1$ has its maximum magnitude when $\phi = \pm\pi/2$. If the frequency is constant, but the arrival angle changes from ϕ to $\phi + \Delta\phi$, then the phase-shift change is

$$\Delta\theta_2 = \frac{2\pi d}{\lambda}[\sin(\phi + \Delta\phi) - \sin\phi]. \tag{48}$$

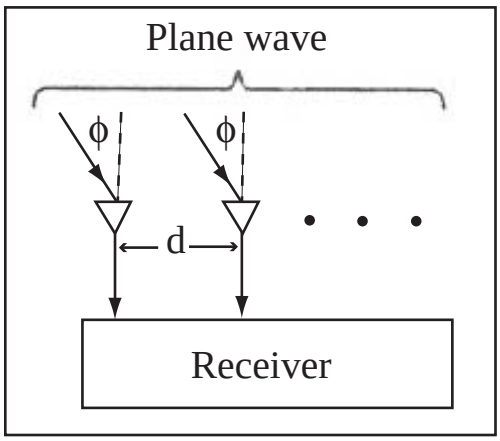

Figure 18. Two array antennas receiving plane wave.

The phase shift experienced by the two antennas is the same for a frequency change Δf as it is for an arrival-angle change $\Delta\phi$ if

$$(\Delta\theta_1)\ modulo\ 2\pi = (\Delta\theta_2)\ modulo\ 2\pi. \tag{49}$$

This equation has at least one solution if $\Delta f < f$ and $\pi/2 - \phi$ is sufficiently large. Thus, a change in frequency due to frequency hopping may be processed as if an equivalent change occurred in the arrival angle.

As an example, suppose that an adaptive system creates an approximate spatial null in the direction $\phi = 0.4\pi = 72^\circ$ of an interference signal when the carrier frequency of a desired frequency-hopping signal is f. If the carrier frequency changes to $f + \Delta f$ such that $\Delta f/f = 0.05$, then equations 46, 48, and 49 indicate that an equivalent arrival-angle change for the interference is $\Delta\phi = 15.0^\circ$. If $\Delta\phi$ exceeds the angular width of the original null after the carrier frequency of the desired signal hops to $f + \Delta f$, then the interference is not immediately cancelled, but further adaptation is required to again establish a spatial null. If equation 49 is satisfied, then equation 48 implies that the performance degradation of an adaptive algorithm due to a large fractional bandwidth increases with d/λ and the arrival angle ϕ. Since the angular width of a null tends to decrease as the interference power increases, this degradation increases with the interference power.

Figures 19 and 20 show the results for typical simulation trials in which $d = \lambda$, a single interference signal has an arrival direction $\phi = 85^\circ$ and an $ISR = 10$ dB, and the other parameter values of table 1 are used. The hopping bandwidth is equal to 75 MHz and 300 MHz in figures 19 and 20, respectively. One observes that the larger fractional bandwidth causes an increase in the SINR variation, as expected from the preceding analysis.

The angular widths of nulls and the main beam can be increased by decreasing the separation among antennas in the array or inserting adaptive filters behind each antenna, but then the resolution of signals from similar directions may be inadequate. The restriction of the frequency change after a hop to a small value maintains the interference near the center of a spatial null. However, the variety of hopping patterns is reduced, which diminishes the resistance to frequency-selective fading and multitone jamming. Two viable methods of frequency compensation for a large fractional bandwidth are spectral processing and anticipative processing.

Spectral processing is based on dividing the hopping band into a number of spectral regions and adapting independently when the carrier frequency is in one of the regions. A separate weight vector associated with each spectral region is stored in a memory device within the adaptive filter. At the end of the signal dwell interval in a specific spectral region, the weight vector generated by the adaptive filter is transferred to the memory, where it replaces a previously stored weight vector associated with this spectral region. Then the weight vector in the adaptive filter is set equal to the previously stored weight vector

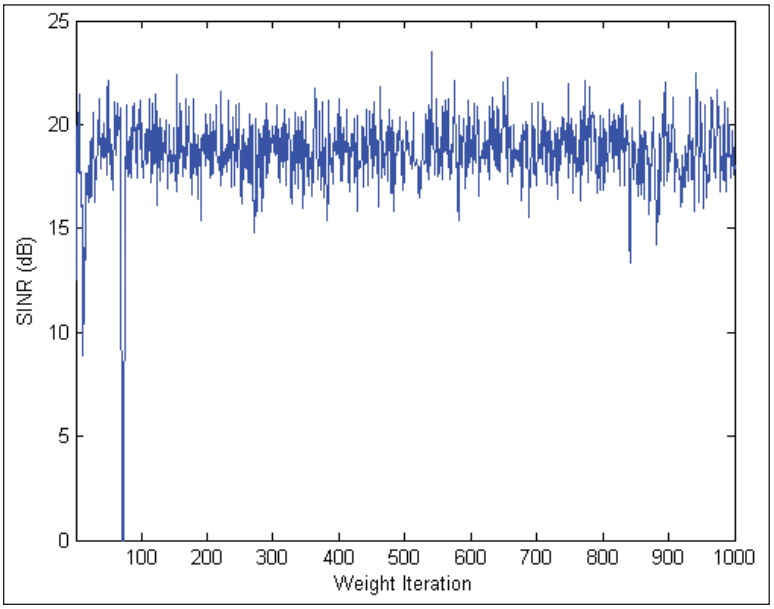

Figure 19. SINR in simulation trial for anticipative maximin algorithm with $W_h = 75$ MHz and one interference signal with $ISR = 10$ dB at 85°.

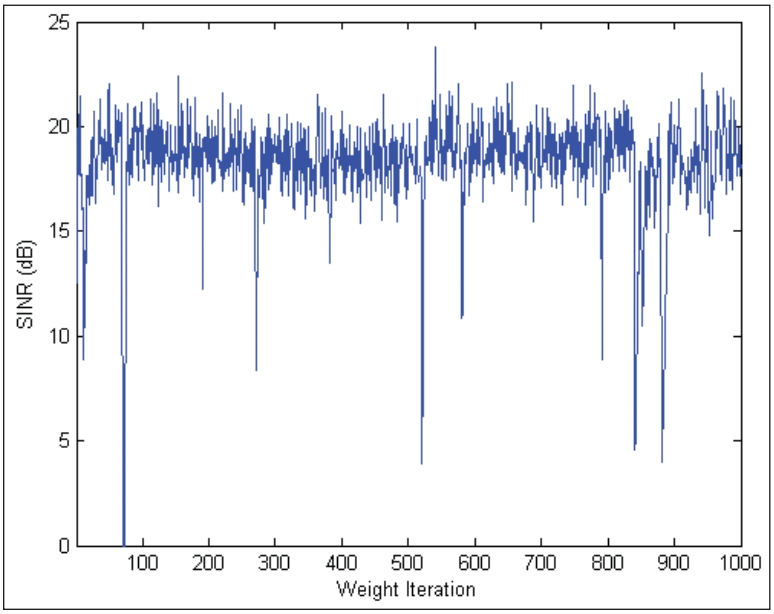

Figure 20. SINR in simulation trial for maximin algorithm with $W_h = 300$ MHz and one interference signal with $ISR = 10$ dB at 85°.

associated with the new spectral region. This weight vector is updated during the dwell interval in the new spectral region. Transfers between the memory device and the weight vector are controlled by the frequency-hopping code generator. As the number of spectral regions N_s increases, there is a proportional increase in the required number of iterations to converge to steady state because only $1/N_s$ of frequency-hopping pulses are associated with each region. For example, the typical performance displayed in figure 20 can be improved to that displayed in figure 19 if spectral processing is used with $N_s = 4$ and a convergence rate reduced by a factor of 4 is acceptable.

When the anticipative maximin algorithm is used, the anticipative branch sequence in figure 7 is a function of the interference signal in the next frequency channel and, hence, frequency compensation can occur for this interference signal. However, the main branch sequence applied to the anticipative processor is extracted from the current signal channel, and thus frequency compensation for the desired signal is not possible. Despite this partial lack of frequency compensation, the anticipative maximin algorithm is useful for wideband frequency hopping in a partial-band interference environment. An alternative anticipative algorithm that omits the factor $\mathbf{c}_x/\hat{p}_x(k)$ in equation 42 is found to be counterproductive because the absence of the desired signal in the anticipative processing causes a distortion of the array gain pattern.

Figure 21 depicts the results of a typical simulation trial under the same conditions as figure 20 except that the anticipative maximin algorithm is used. The figure displays an improved convergence and is very similar to what is obtained for the anticipative maximin algorithm under the same conditions as figure 19. The results of simulation experiments with 20 trials and $ISR = 10$ dB per channel are listed in table 5. They confirm that the anticipative maximin algorithm is an effective method of frequency compensation. The performance when the anticipative maximin algorithm is used and $W_h = 300$ MHz exceeds the performance when the maximin algorithm is used and $W_h = 75$ MHz.

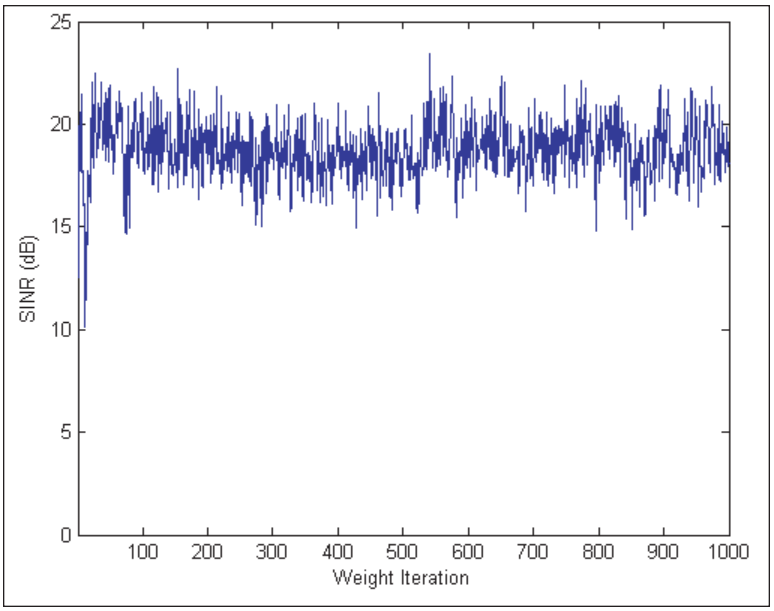

Figure 21. SINR in simulation trial for anticipative maximin algorithm with $W_h = 300$ MHz and one interference signal with $ISR = 10$ dB at 85°.

Table 5. Simulation results for $ISR = 10$ dB per channel.

Interfer. Angles	W_h(MHz)	d/λ	**Anticip.**	**Average SINR(dB)**	**Stand. dev.(dB)**	**Final SINR(dB)**	**Final St. dev.(dB)**
85°	75	1	no	18.53	2.17	18.81	1.43
85°	75	1	yes	19.00	1.77	19.04	1.31
85°	300	1	no	18.29	2.63	18.54	2.07
85°	300	1	yes	18.93	1.80	18.99	1.35
40°, −10°	75	1.5	no	16.03	5.33	15.73	3.19
40°, −10°	75	1.5	yes	17.05	2.42	16.40	1.36
40°, −10°	300	1.5	no	15.60	5.82	15.22	4.09
40°, −10°	300	1.5	yes	16.95	2.59	16.41	1.64

7. Conclusions

The anticipative maximin algorithm uses both spatial and temporal techniques to provide a frequency-hopping system with a high degree of protection against strong interference that cannot be accommodated by the frequency hopping and error-control coding alone. Major advantages of the algorithm are that it does not require training sequences, decision-directed adaptation, directional information, or elaborate computations such as eigenanalysis. The algorithm provides frequency compensation when the hopping spans a wide spectral band. Compared with the maximin algorithm, the anticipative maximin algorithm provides additional protection against interference at the cost of additional hardware and nearly a doubling of the computational requirements.

References

[1] Acar, L.; Compton, L. T. The Performance of an LMS Adaptive Array with Frequency-Hopped Signals. *IEEE Trans. Aerosp. Electron. Syst.* **1985**, *21*, 360–371.

[2] Torrieri, D.; Bakhru, K. Frequency compensation in an adaptive antenna system for frequency-hopping communications. *IEEE Trans. Aerosp. Electron. Syst.* **1987**, *23*, 448–467.

[3] Torrieri, D.; Bakhru, K. Blind adaptation using maximin algorithm. *Proc. Asilomar Conf. on Signals, Systems, and Computers*, **1993**, 638–642.

[4] Kamiya, Y.; Besson, O. Interference Rejection for Frequency-Hopping Communication Systems Using a Constant Power Algorithm. *IEEE Trans. Commun.* **2003**, *51*, 627–633.

[5] Torrieri, D.; Bakhru, K. An Anticipative Adaptive Array for Frequency-Hopping Communications. *IEEE Trans. Aerosp. Electron. Syst.* **1988**, *24*, 449–456.

[6] Torrieri, D. *Principles of Spread-Spectrum Communication Systems*; Springer: Boston, MA, 2005.

[7] Haykin, S. *Adaptive Filter Theory, 4th ed.*; Prentice-Hall: Upper Saddle River, NJ, 2002.

[8] Sayed, A. H. *Fundamentals of Adaptive Filtering*; Wiley: Hoboken, NJ, 2003.

[9] Van Trees, H. L. *Optimum Array Processing*; Wiley: Hoboken, NJ, 2002.

[10] Porat, B. *A Course in Digital Signal Processing*; Wiley: New York, NY, 1997.

[11] Torrieri, D.; Bakhru, K. Adaptive-Array Algorithm for Interference Suppression Prior to Acquisition of Direct-Sequence Signal. *Proc. IEEE MILCOM Conf.*, **2005**.

A. Stationary Stochastic Processes

Consider a real-valued stochastic process $n(t)$ that is a zero-mean, wide-sense stationary process with autocorrelation

$$R_n(\tau) = E[n(t)n(t+\tau)] \tag{A.1}$$

where $E[x]$ denotes the expected value of x. The Hilbert transform of this process is the stochastic process defined as

$$\hat{n}(t) = H[n(t)] = \frac{1}{\pi}\int_{-\infty}^{\infty}\frac{n(u)}{t-u}du \tag{A.2}$$

where it is assumed that the Cauchy principal value of the integral exists for almost every sample function of $n(t)$. This equation indicates that $\hat{n}(t)$ is a zero-mean stochastic process. The zero-mean processes $n(t)$ and $\hat{n}(t)$ are *jointly wide-sense stationary* if their correlation and cross-correlation functions are not functions of t. Assuming that the expectation and the integration are interchangeable, equations A-2 and A-1 imply that the cross-correlation is

$$R_{n\hat{n}}(\tau) = E[n(t)\hat{n}(t+\tau)] = \frac{1}{\pi}\int_{-\infty}^{\infty}\frac{R_n(u)}{\tau-u}du = \hat{R}_n(\tau). \tag{A.3}$$

Since equation A-2 has the form of the convolution of the sample function $n(t)$ with $1/\pi t$, $\hat{n}(t)$ results from passing $n(t)$ through a linear filter with an impulse response equal to $1/\pi t$. The transfer function of the filter is given by the Fourier transform

$$\mathcal{F}\left\{\frac{1}{\pi t}\right\} = \int_{-\infty}^{\infty}\frac{\exp(-j2\pi ft)}{\pi t}\,dt \tag{A.4}$$

where $j = \sqrt{-1}$. This integral can be rigorously evaluated by using contour integration. Alternatively, we observe that since $1/t$ is an odd function,

$$\begin{aligned}\mathcal{F}\left\{\frac{1}{\pi t}\right\} &= -2j\int_{0}^{\infty}\frac{\sin 2\pi ft}{\pi t}\,dt \\ &= -j\,\mathrm{sgn}(f)\end{aligned} \tag{A.5}$$

where $\mathrm{sgn}(f)$ is the *signum function* defined by

$$\mathrm{sgn}(f) = \begin{cases} 1, & f > 0 \\ 0, & f = 0 \\ -1, & f < 0 \end{cases} \tag{A.6}$$

Let $G(f) = \mathcal{F}\{n(t)\}$, and let $\hat{G}(f) = \mathcal{F}\{\hat{n}(t)\}$. Equations A-2 and A-5 and the convolution theorem imply that

$$\hat{G}(f) = -j \operatorname{sgn}(f) G(f). \tag{A.7}$$

Because $H[\hat{n}(t)]$ results from passing $n(t)$ through two successive filters, each with transfer function $-j \operatorname{sgn}(f)$,

$$H[\hat{n}(t)] = -n(t) \tag{A.8}$$

provided that $G(0) = 0$. Using equations A-2, A-1, successive changes in integration variables, and equation A-8, we obtain the autocorrelation of $\hat{n}(t)$:

$$\begin{aligned} R_{\hat{n}}(\tau) &= E[\hat{n}(t)\hat{n}(t+\tau)] \\ &= \frac{1}{\pi^2} \int_{-\infty}^{\infty} \frac{1}{t-u} du \int_{-\infty}^{\infty} \frac{R_n(v-u)}{t+\tau-v} dv = \frac{1}{\pi} \int_{-\infty}^{\infty} \frac{\hat{R}_n(t+\tau-u)}{t-u} du = -H[\hat{R}_n(\tau)] \\ &= R_n(\tau). \end{aligned} \tag{A.9}$$

Equations A-1, A-3, and A-9 indicate that $n(t)$ and $\hat{n}(t)$ are jointly wide-sense stationary.

The *analytic signal* associated with $n(t)$ is the zero-mean process defined by

$$n_a(t) = n(t) + j\hat{n}(t). \tag{A.10}$$

The autocorrelation of the analytic signal is defined as

$$R_a(\tau) = E[n_a^*(t) n_a(t+\tau)] \tag{A.11}$$

where the asterisk denotes the complex conjugate. Using equations A-1, A-3, and A-9 to A-11, we obtain

$$R_a(\tau) = 2R_n(\tau) + 2j\hat{R}_n(\tau) \tag{A.12}$$

which establishes the wide-sense stationarity of the analytic signal. Since equation A-1 indicates that $R_n(\tau)$ is an even function, equation A-3 yields

$$R_{n\hat{n}}(0) = \hat{R}_n(0) = 0 \tag{A.13}$$

which indicates that $n(t)$ and $\hat{n}(t)$ are uncorrelated. Equations A-9, A-12, and A-13 yield

$$R_{\hat{n}}(0) = R_n(0) = 1/2 R_a(0) \tag{A.14}$$

The *complex envelope* of $n(t)$ or the *equivalent lowpass representation* of $n(t)$ is the zero-mean stochastic process defined by

$$n_l(t) = n_a(t) \exp(-j2\pi f_c t) \tag{A.15}$$

where f_c is an arbitrary frequency usually chosen as the center or carrier frequency of $n(t)$. The complex envelope can be decomposed as

$$n_l(t) = n_c(t) + j n_s(t) \tag{A.16}$$

where $n_c(t)$ and $n_s(t)$ are real-valued, zero-mean stochastic processes. Equations A-10, A-15, and A-16 imply that

$$\begin{aligned} n(t) &= \mathrm{Re}[n_l(t) \exp(j2\pi f_c t)] \\ &= n_c(t) \cos(2\pi f_c t) - n_s(t) \sin(2\pi f_c t). \end{aligned} \tag{A.17}$$

Substituting equations A-10 and A-16 into A-15 we find that

$$n_c(t) = n(t) \cos(2\pi f_c t) + \hat{n}(t) \sin(2\pi f_c t) \tag{A.18}$$

$$n_s(t) = \hat{n}(t) \cos(2\pi f_c t) - n(t) \sin(2\pi f_c t) \tag{A.19}$$

The *autocorrelations* of $n_c(t)$ and $n_s(t)$ are defined by

$$R_c(\tau) = E[n_c(t) n_c(t+\tau)]. \tag{A.20}$$

and

$$R_s(\tau) = E[n_s(t) n_s(t+\tau)] \tag{A.21}$$

Using equations A-18 and A-19 and then equations A-1, A-3, and A-9, and trigonometric identities, we obtain

$$R_c(\tau) = R_s(\tau) = R_n(\tau) \cos(2\pi f_c \tau) + \hat{R}_n(\tau) \sin(2\pi f_c \tau). \tag{A.22}$$

which shows explicitly that if $n(t)$ is wide-sense stationary, then $n_c(t)$ and $n_s(t)$ are wide-sense stationary with the same autocorrelation function. The variances of $n(t)$, $n_c(t)$, and $n_s(t)$ are all equal because

$$R_c(0) = R_s(0) = R_n(0). \tag{A.23}$$

A derivation similar to that of equation A-22 gives the cross-correlation

$$R_{cs}(\tau) = E[n_c(t)n_s(t+\tau)] = \hat{R}_n(\tau)\cos(2\pi f_c\tau) - R_n(\tau)\sin(2\pi f_c\tau). \tag{A.24}$$

Equations A-22 and A-24 indicate that $n_c(t)$ and $n_s(t)$ are jointly wide-sense stationary, which then implies that

$$R_{sc}(\tau) = E[n_s(t)n_c(t+\tau)] = R_{cs}(-\tau) \tag{A.25}$$

Equations A-13 and A-24 give

$$R_{cs}(0) = 0. \tag{A.26}$$

which implies that $n_c(t)$ and $n_s(t)$ are uncorrelated.

Since $n(t)$ is wide-sense stationary, $R_n(-\tau) = R_n(\tau)$. It then follows from equation A-3 and a change of the integration variable that $\widehat{R}_n(-\tau) = -\widehat{R}_n(\tau)$. Combining these equations with equation A-24 yields $R_{cs}(-\tau) = -R_{cs}(\tau)$. This equation and equation A-25 indicate that

$$R_{cs}(\tau) = -R_{sc}(\tau) \tag{A.27}$$

Equations A-16, A-23, and A-27 imply that

$$E[n_l(t)n_l(t+\tau)] = 0. \tag{A.28}$$

A complex-valued, zero-mean stochastic process that satisfies this equation is called a *circularly symmetric* process. Thus, the complex envelope of a zero-mean, wide-sense stationary process is a circularly symmetric process.

B. Convergence of Maximin Algorithm

B.1 Optimal Weight Vector

The SINR at the adaptive-filter output is

$$\rho = \frac{\mathbf{w}^H \mathbf{R}_s \mathbf{w}}{\mathbf{w}^H \mathbf{R}_n \mathbf{w}}. \tag{B.1}$$

We seek to find the weight vector that maximizes the SINR. Since the $N \times N$ correlation matrices $\mathbf{R}_s = E[\mathbf{s}(i)\mathbf{s}^H(i)]$ and $\mathbf{R}_n = E[\mathbf{n}(i)\mathbf{n}^H(i)]$ are Hermitian and nonnegative definite, they have complete sets of orthonormal eigenvectors with nonnegative real-valued eigenvalues. The noise power is assumed to be positive, which implies that $\mathbf{R}_n$ is positive definite and has positive eigenvalues. The spectral theorem of linear algebra indicates that $\mathbf{R}_n$ can be expressed as

$$\mathbf{R}_n = \sum_{i=1}^{N} \lambda_i \mathbf{e}_i \mathbf{e}_i^H \tag{B.2}$$

where λ_i is an eigenvalue and $\mathbf{e}_i$ is the associated eigenvector.

To derive the weight vector that maximizes the SINR with no restriction on $\mathbf{R}_s$, we define the Hermitian matrix

$$\mathbf{A} = \sum_{i=1}^{L} \sqrt{\lambda_i} \mathbf{e}_i \mathbf{e}_i^H \tag{B.3}$$

where the positive square root is used. A direct calculation verifies that

$$\mathbf{R}_n = \mathbf{A}^2. \tag{B.4}$$

Since the spectral theorem implies that $I = \sum_{i=1}^{N} \mathbf{e}_i \mathbf{e}_i^H$, the inverse of $\mathbf{A}$ is

$$\mathbf{A}^{-1} = \sum_{i=1}^{L} \frac{1}{\sqrt{\lambda_i}} \mathbf{e}_i \mathbf{e}_i^H. \tag{B.5}$$

The matrix $\mathbf{A}$ specifies an invertible transformation of $\mathbf{w}$ into the vector

$$\mathbf{v} = \mathbf{A}\mathbf{w}. \tag{B.6}$$

We define the Hermitian matrix

$$\mathbf{C} = \mathbf{A}^{-1} \mathbf{R}_s \mathbf{A}^{-1}. \tag{B.7}$$

Then equations B-1, B-4, B-6, and B-7 indicate that the SINR can be expressed as a *Rayleigh quotient*:

$$\rho = \frac{\mathbf{v}^H \mathbf{C}\ \mathbf{v}}{\|\ \mathbf{v}\ \|^2}. \tag{B.8}$$

The original maximization problem has been reduced to the maximization of a Rayleigh quotient. An alternative derivation of a Rayleigh quotient can be obtained from the Cholesky decomposition $\mathbf{R}_n = \mathbf{B}\mathbf{B}^H$, where $\mathbf{B}$ is lower triangular with positive diagonal elements.

Let $\ell_1 \geq \ell_2 \geq \cdots \geq \ell_N$ and $\mathbf{u}_1, \cdots, \mathbf{u}_N$ denote the eigenvalues and corresponding orthonormal eigenvectors of $\mathbf{C}$. If $\mathbf{v}$ is expanded as

$$\mathbf{v} = \sum_{i=1}^{N} b_i \mathbf{u}_i \tag{B.9}$$

where the $\{b_i\}$ are coefficients, then

$$\mathbf{v}^H \mathbf{C}\ \mathbf{v} = \sum_{i=1}^{N} |b_i|^2 \ell_i \leq \ell_1 \sum_{i=1}^{N} |b_i|^2 = \ell_{max}\ \|\ \mathbf{v}\ \|^2 \tag{B.10}$$

where $\ell_{max} = \ell_1$ is the largest eigenvalue. Therefore, $\rho \leq \ell_{max}$. Direct substitution indicates that ρ is maximized by $\mathbf{v} = \eta\mathbf{u}$, where $\mathbf{u}$ is an eigenvector of $\mathbf{C}$ associated with eigenvalue l_{max}, and η is an arbitrary constant. Thus, the maximum value of ρ is

$$\rho_0 = \ell_{max}. \tag{B.11}$$

From equation B-6 with $\mathbf{v} = \eta\mathbf{u}$, it follows that an *optimal weight vector that maximizes the SINR* is

$$\mathbf{w}_0 = \eta\mathbf{A}^{-1}\mathbf{u}. \tag{B.12}$$

The purpose of an adaptive-array algorithm is to adjust the weight vector to converge to the optimal value, which is given by equation B-12 when the maximization of the SINR is the performance criterion.

An alternative means of evaluating $\mathbf{w}_0$ follows from observing that $\rho_0\mathbf{w}_0 = \rho_0\mathbf{A}^{-1}\mathbf{u} = \mathbf{A}^{-1}\mathbf{C}\mathbf{u} = \mathbf{A}^{-2}\mathbf{R}_s\mathbf{A}^{-1}\mathbf{u} = \mathbf{R}_n^{-1}\mathbf{R}_s\mathbf{w}_0$. Therefore, $\mathbf{w}_0$ is an eigenvector of $\mathbf{R}_n^{-1}\mathbf{R}_s$ with eigenvalue ρ_0. A substitution of $\mathbf{R}_n^{-1}\mathbf{R}_s\mathbf{w}_0 = \rho_0\mathbf{w}_0$ into equation B-1 yields $\rho = \rho_0$, which proves that ρ_0 is the largest eigenvalue associated with $\mathbf{R}_n^{-1}\mathbf{R}_s$.

Let $s(i)$ denote the component of $\mathbf{s}(i)$ derived from a fixed reference antenna. In practical spread-spectrum systems, the despread desired signal is sufficiently narrowband that its copies in all the branches are nearly aligned in time. Therefore,

$$\mathbf{s}(i) = s(i)\mathbf{s}_0 \tag{B.13}$$

where $\mathbf{s}_0$ is a *steering vector* of complex numbers that represent the relative amplitudes and phase shifts at the antenna outputs. When equation B-13 is a valid representation, equation B-12 can be simplified. The substitution of equation B-13 into $\mathbf{R}_s = E[\mathbf{s}(i)\mathbf{s}^H(i)]$ yields

$$\mathbf{R}_s = p_{si}\mathbf{s}_0\mathbf{s}_0^H \tag{B.14}$$

where

$$p_{si} = E[|s(i)|^2]. \tag{B.15}$$

The substitution into equation B-14 into B-7 indicates that $\mathbf{C}$ may be factored:

$$\mathbf{C} = p_{si}\mathbf{f}\mathbf{f}^H \tag{B.16}$$

where $\mathbf{f} = \mathbf{A}^{-1}\mathbf{s}_0$. The factorization explicitly shows that $\mathbf{C}$ is a rank-one matrix, which has only one nonzero eigenvalue. By direct substitution, it is found that the eigenvector associated with the nonzero eigenvalue is

$$\mathbf{u} = \mathbf{f} = \mathbf{A}^{-1}\mathbf{s}_0 \tag{B.17}$$

and the eigenvalue is

$$\ell_{max} = p_{si} \parallel \mathbf{f} \parallel^2 . \tag{B.18}$$

Substituting equation B-17 into B-12 and then observing that $\mathbf{R}_n^{-1} = \mathbf{A}^{-2}$, we obtain the *optimal weight vector*:

$$\mathbf{w}_0 = \eta\mathbf{R}_n^{-1}\mathbf{s}_0 \tag{B.19}$$

where η is an arbitrary constant. The maximum value of the SINR, obtained from equations B-11, B-18, and B-17 is

$$\rho_0 = p_{si}\mathbf{s}_0^H\mathbf{R}_n^{-1}\mathbf{s}_0. \tag{B.20}$$

B.2 Convergence of Mean Weight Vector

The highly nonlinear nature of the maximin algorithm precludes a completely rigorous convergence analysis. However, with enough approximations, the convergence of the mean weight vector to the optimal weight vector can be demonstrated, and bounds on the adaptation constant can be derived. We assume that the interference is wide-sense stationary and m is large enough that equation 40 gives

$$\hat{t}(k) \approx r = E[\hat{t}(k)] = E[\| \hat{\mathbf{n}}(i) \|^2] \approx E[\| \mathbf{n}(i) \|^2] = tr(\mathbf{R}_n) \tag{B.21}$$

where $tr(\ )$ denotes the trace. We assume that after a number of algorithm iterations k_0, $\hat{p}_x(k)/\hat{p}_n(k) \approx \rho + 1 \approx \rho_0 + 1$. Using these assumptions in equation 22 and 23, the maximin algorithm is approximated by

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \frac{\alpha}{r}\left[\frac{\mathbf{c}_x(k)}{(\rho_0+1)} - \mathbf{c}_n(k)\right], \qquad k \geq k_0. \tag{B.22}$$

Since $\mathbf{w}(k)$ does not depend on $\mathbf{x}(i)$ and $\widehat{\mathbf{n}}(i)$ for $i \geq km+1$, we make the approximation that $\mathbf{w}(k)$ is statistically independent of $\mathbf{x}(i)$ and $\mathbf{n}(i)$ for $i \geq km+1$. We obtain from equations 33, 32, 16, and 12 that

$$E[\mathbf{c}_x(k)] = E[\mathbf{x}(i)y_r(i)] = \frac{1}{2}\mathbf{R}_x E[\mathbf{w}(k)]. \tag{B.23}$$

Similarly, equations 38, 37, 17, 7, and 3 yield

$$E[\mathbf{c}_n(k)] = E[\hat{\mathbf{n}}(i)\hat{y}_{nr}(i)] \approx \frac{1}{2}\mathbf{R}_n E[\mathbf{w}(k)]. \tag{B.24}$$

Taking the expected value of both sides of equation B-22, substituting equations B-23, B-24, and 12 and simplifying algebraically, we obtain the approximate recursive equation for the mean weight vector:

$$E[\mathbf{w}(k+1)] = \left[\mathbf{I} - \frac{\alpha}{2r(\rho_0+1)}\mathbf{D}\right] E[\mathbf{w}(k)], \qquad k \geq k_0 \tag{B.25}$$

where

$$\mathbf{D} = \rho_0\mathbf{R}_n - \mathbf{R}_s = \rho_0\mathbf{R}_n - p_{si}\mathbf{s}_0\mathbf{s}_0^H \ . \tag{B.26}$$

A straightforward calculation using equationsB-26 and B-20 yields

$$\mathbf{D}\ \mathbf{R}_n^{-1}\mathbf{s}_0 = \mathbf{0} \tag{B.27}$$

which indicates that $\mathbf{w}_0 = \mathbf{R}_n^{-1}\mathbf{s}_0$ is an eigenvector of $\mathbf{D}$, and the corresponding eigenvalue is 0. Since $\mathbf{D}$ is Hermitian, it has a complete set of N orthogonal eigenvectors, one of which is $\mathbf{R}_n^{-1}\mathbf{s}_0$. Since $\rho \leq \rho_0$, equation B-1 implies that $\mathbf{w}^H\mathbf{R}_s\mathbf{w} \leq \rho_0\mathbf{w}^H\mathbf{R}_n\mathbf{w}$. Consequently, $\mathbf{w}^H\mathbf{D}\ \mathbf{w} \geq 0$, which proves that $\mathbf{D}$ is positive semidefinite and, hence, has N nonnegative eigenvalues. Since only $\mathbf{w} = \mathbf{w}_0$ gives $\rho = \rho_0$, one of these eigenvalues is zero, and the other $N-1$ eigenvalues are positive.

We make the decomposition

$$E[\mathbf{w}(k)] = \eta(k)\mathbf{R}_n^{-1}\mathbf{s}_0 + \sum_{i=2}^{N} a_i(k)\mathbf{e}_i \tag{B.28}$$

where each $a_i(k)$ and $\eta(k)$ are scalar functions and $\mathbf{e}_i$ is one of the $N-1$ eigenvectors orthogonal to $\mathbf{R}_n^{-1}\mathbf{s}_0$. Substituting this equation into equation B-25 and using the orthogonality of the eigenvectors, we obtain

$$\eta(k+1) = \eta(k) = \eta(k_0), \qquad k \geq k_0 \tag{B.29}$$

$$a_i(k+1) = \left[1 - \frac{\alpha\lambda_i}{2r(\rho_0+1)}\right] a_i(k), \quad 2 \leq i \leq N, \qquad k \geq k_0 \tag{B.30}$$

where λ_i is the eigenvalue corresponding to $\mathbf{e}_i$. Assuming that $\eta(k_0) \neq 0$, equations B-19, B-29, and B-30 indicate that $E[\mathbf{w}(k)] \longrightarrow \mathbf{w}_0$ as $k \to \infty$ if and only if each $a_i(k) \to 0$. The solution to equation B-30 is

$$a_i(k) = \left[1 - \frac{\alpha\lambda_i}{2r(\rho_0+1)}\right]^{k-k_0} a_i(k_0), \quad 2 \leq i \leq N, \qquad k \geq k_0. \tag{B.31}$$

This equation indicates that $a_i(k) \longrightarrow 0$, $2 \leq i \leq N$, as $k \to \infty$ if and only if

$$\left|1 - \frac{\alpha\lambda_i}{2r(\rho_0+1)}\right| < 1\ , \quad 2 \leq i \leq N. \tag{B.32}$$

This inequality implies that the necessary and sufficient condition for the convergence of the mean weight vector is

$$0 < \alpha < \frac{4r(\rho_0+1)}{\lambda_{max}} \tag{B.33}$$

where λ_{max} is the largest eigenvalue of $\mathbf{D}$.

Since the sum of the eigenvalues of a square matrix is equal to its trace,

$$\lambda_{max} \leq \sum_{i=1}^{N} \lambda_i = tr(\mathbf{D}) = \rho_0 r - tr(\mathbf{R}_s) \leq \rho_0 r. \tag{B.34}$$

Substituting this bound into equation B-33 and simplifying the result, we obtain

$$0 < \alpha < 4 \tag{B.35}$$

as a sufficient (but not necessary) condition for the convergence of the mean weight vector to the optimal weight vector. Although this inequality must be regarded as an approximation because of the approximations used in its derivation, it gives at least rough guidance in the selection of the adaptation constant. The fact that the upper bound is numerical and does not depend on environmental parameters provides support for the choice of equation 23 as the adaptation sequence.

C. Alternative Maximin Algorithm

The alternative maximin algorithm is derived in the same manner as the maximin algorithm except that the circular symmetry of complex envelopes is not exploited. As shown in the derivation of equation 15, the method of steepest descent yields

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mu_0(k)[\rho(k)+1]\left[\frac{\mathbf{R}_x\mathbf{w}(k)}{p_x(k)} - \frac{\mathbf{R}_n\mathbf{w}(k)}{p_n(k)}\right]. \tag{C.1}$$

Let $\hat{p}_x(k)$, $\hat{p}_n(k)$, and $\widehat{\rho}(k)$ denote estimates of $p_x(k)$, $p_n(k)$, and $\rho(k)$, respectively, following weight iteration k. The *adaptation sequence* is defined as $\alpha(k) = \mu_0(k)[\widehat{\rho}(k)+1]$. Let $\mathbf{c}_x(k)$ and $\mathbf{c}_n(k)$ denote estimates following iteration k of the input correlation vector $E[\mathbf{x}(i)y^*(i)]$ and the noise correlation vector $E[\mathbf{n}(i)y_n^*(i)]$, respectively. If $\mathbf{w}(k)$ is modeled as deterministic (nonrandom) between weight iterations, then $\mathbf{R}_x\mathbf{w}(k) = E[\mathbf{x}(i)y^*(i)]$ and $\mathbf{R}_n\mathbf{w}(k) = E[\mathbf{n}(i)y_n^*(i)]$. Substituting these estimates and relations into equation C-1, we obtain the *alternative maximin algorithm:*

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \alpha(k)\left[\frac{\mathbf{c}_x(k)}{\hat{p}_x(k)} - \frac{\mathbf{c}_n(k)}{\hat{p}_n(k)}\right], \qquad k \geq 0 \tag{C.2}$$

where $\mathbf{w}(0)$ is the deterministic initial weight vector. The adaptation sequence $\alpha(k)$ should be chosen so that $E[\mathbf{w}(k)]$ converges to a nearly optimal steady-state value. It is also intuitively plausible that $\alpha(k)$ should decrease rapidly as $E[\mathbf{w}(k)]$ converges. A suitable candidate is

$$\alpha(k) = \alpha\frac{\hat{p}_n(k)}{\hat{t}(k)} \tag{C.3}$$

where $\hat{t}(k)$ is an estimate of the total interference and noise power entering the array, and α is the *adaptation constant.*

The vectors applied to the adaptive filter are $\mathbf{x}(i)$ and $\hat{\mathbf{n}}(i)$, and m samples are taken per weight iteration. The adaptive filter produces the output

$$y(i) = \mathbf{w}^H(k)\mathbf{x}(i), \quad i = km+1, \cdots, (k+1)m \tag{C.4}$$

where sample i is taken after weight iteration k. This output is applied to the demodulator and is used in the estimators

$$\mathbf{c}_x(k) = \frac{1}{m}\sum_{i=km+1}^{(k+1)m}\mathbf{x}(i)y^*(i), \qquad k \geq 0 \tag{C.5}$$

and

$$\hat{p}_x(k) = \frac{1}{m} \sum_{i=km+1}^{(k+1)m} |y(i)|^2, \qquad k \geq 0 \tag{C.6}$$

which are unbiased when the desired signal and the interference are wide-sense stationary processes.

The adaptive filter generates an estimate of $y_n(i) = \mathbf{w}^H(k)\mathbf{n}(i)$ given by

$$\hat{y}_n(i) = \mathbf{w}^H(k)\hat{\mathbf{n}}(i), \quad i = km+1, \cdots, (k+1)m. \tag{C.7}$$

The critical requirement of $\hat{\mathbf{n}}(i)$ is that it have the same second-order statistics as $\mathbf{n}(i)$, the undesired component of $\mathbf{x}(i)$. If the spectrum of an interference signal overlaps the signal and monitor channels, then the vector

$$\mathbf{n}_e(i) = \hat{\mathbf{n}}(i) \exp(-j2\pi f_o i T_s) \tag{C.8}$$

provides an estimate of the signal-filter output $\mathbf{n}(i)$. Treating the weight vector as a constant and observing that $E[\mathbf{n}(i)\mathbf{n}^H(i)] \approx E[\mathbf{n}_e(i)\mathbf{n}_e^H(i)] = E[\hat{\mathbf{n}}(i)\hat{\mathbf{n}}^H(i)]$, we obtain

$$E[\hat{\mathbf{n}}(i)\hat{y}_n^*(i)] \approx E[\mathbf{n}(i)y_n^*(i)]. \tag{C.9}$$

Therefore, an estimator of the interference-plus-noise correlation at weight iteration k is

$$\mathbf{c}_n(k) = \frac{1}{m} \sum_{i=km+1}^{(k+1)m} \hat{\mathbf{n}}(i)\hat{y}_n^*(i), \qquad k \geq 0. \tag{C.10}$$

Similarly, an estimator proportional to the interference-plus-noise output power is

$$\hat{p}_n(k) = \frac{1}{m} \sum_{i=km+1}^{(k+1)m} |\hat{y}_n(i)|^2, \qquad k \geq 0. \tag{C.11}$$

A recursive estimator of the total interference-plus-noise power entering the array is the same as it is in the maximin algorithm:

$$\hat{t}(k) = \begin{cases} \mu\hat{t}(k-1) + \frac{1-\mu}{m} \sum_{i=km+1}^{(k+1)m} \| \hat{\mathbf{n}}(i) \|^2, & k \geq 1 \\ \frac{1}{m} \sum_{i=1}^{m} \| \hat{\mathbf{n}}(i) \|^2, & k = 0 \end{cases} \tag{C.12}$$

where $\| \mathbf{v} \|$ denotes the Euclidean norm of $\mathbf{v}$, μ is the *memory factor*, and $0 \leq \mu \leq 1$.

The convergence analysis for the alternative algorithm is nearly the same as that for the maximin algorithm, which is derived in appendix B. It is found that for the alternative algorithm,

$$0 < \alpha < 2 \tag{C.13}$$

is a sufficient (but not necessary) condition for the convergence of the mean weight vector to the optimal weight vector given by equation B-19.

Let $\mathbf{u}(k)$ denote the second term on the right-hand side of equation C-2. A straightforward calculation using the preceding equations indicates that $\mathbf{w}^H(k)\mathbf{u}(k) = \mathbf{0}$. Consequently, $\| \mathbf{w}(k+1) \|^2 = \| \mathbf{w}(k) + \mathbf{u}(k) \|^2 \geq \| \mathbf{w}(k) \|^2$, which implies that the norm of the weight vector increases monotonically. Thus, the alternative algorithm requires some type of weight normalization or regularization. Even without including this normalization, the alternative algorithm requires almost twice as much computation as the maximin algorithm. Simulation experiments indicate that the two algorithms provide nearly the same overall performance. Therefore, the maximin algorithm is clearly preferable to the alternative maximin algorithm.

INTENTIONALLY LEFT BLANK.

Distribution

ADMNSTR
DEFNS TECHL INFO CTR
ATTN DTIC-OCP (ELECTRONIC COPY)
8725 JOHN J KINGMAN RD STE 0944
FT BELVOIR VA 22060-6218

DARPA
ATTN IXO S WELBY
3701 N FAIRFAX DR
ARLINGTON VA 22203-1714

OFC OF THE SECY OF DEFNS
ATTN ODDRE (R&AT)
THE PENTAGON
WASHINGTON DC 20301-3080

US ARMY TRADOC
BATTLE LAB INTEGRATION & TECHL DIRCTRT
ATTN ATFEDS
10 WHISTLER LANE
FT MONROE VA 23651-5850

SMC/GPA
2420 VELA WAY STE 1866
EL SEGUNDO CA 90245-4659

TECOM
ATTN CSTE-DTC-CL
ABERDEEN PROVING GROUND MD 21005-5057

US ARMY ARDEC
ATTN AMSTA-AR-TD
BLDG 1
PICATINNY ARSENAL NJ 07806-5000

COMMANDING GENERAL
US ARMY AVN & MIS CMND
ATTN AMSAM-RD W C MCCORKLE
REDSTONE ARSENAL AL 35898-5000

US ARMY INFO SYS ENGRG CMND
ATTN AMSEL-IE-TD F JENIA
FT HUACHUCA AZ 85613-5300

US ARMY NATICK RDEC ACTING TECHL DIR
ATTN SBCN-TP P BRANDLER
KANSAS STREET BLDG 78
NATICK MA 01760-5056

US ARMY SIMULATION TRAIN & INSTRMNTN CMND
ATTN AMSTI-CG M MACEDONIA
12350 RESEARCH PARKWAY
ORLANDO FL 32826-3726

CUBIC DEFNS SYS
ATTN K BAKHRU
9323 BALBOA AVE
SAN DIEGO CA 92123

US ARMY RSRCH LAB
ATTN AMSRD-ARL-CI-OK-TP TECHL LIB
T LANDFRIED (2 COPIES)
ABERDEEN PROVING GROUND MD 21005-5066

DIRECTOR
US ARMY RSRCH LAB
ATTN AMSRD-ARL-RO-EV W D BACH
PO BOX 12211
RESEARCH TRIANGLE PARK NC 27709

US ARMY RSRCH LAB
ATTN AMSRD-ARL-CI J GOWENS
ATTN AMSRD-ARL-CI-C D TORRIERI (15 COPIES)
ATTN AMSRD-ARL-CI-OK-T TECHL PUB (2 COPIES)

ATTN AMSRD-ARL-CI-OK-TL TECHL
LIB (2 COPIES)
ATTN AMSRD-ARL-D J M MILLER
ATTN IMNE-ALC-IMS MAIL &
RECORDS MGMT
ADELPHI MD 20783-1197